**Hadoop MapReduce and Hadoop Distributed File System (HDFS)**

**Introduction:**

* Hadoop is an Apache open source framework which is written in Java and allows distributed processing of large datasets across clusters of computers using simple programming models.
* MapReduce is a parallel programming model which runs on Hadoop for writing distributed applications devised at Google for efficient processing of large amounts of data (multi-terabyte datasets), on large clusters (thousands of nodes) of commodity hardware in a reliable, fault-tolerant manner.
* Hadoop Distributed File System is a block-structured file system where each file is divided into blocks of a pre-determined size and these blocks are stored across a cluster of one or several machines. Apache Hadoop HDFS Architecture follows a Master/Slave Architecture, where a cluster comprises of a single NameNode (Master node) and all the other nodes are DataNodes (Slave nodes).

**In Class Exercise:**

Here, first we need to create a JAVA Project and then we need to add external JARs to the project which we have created. So in order to add these libraries we need to right click on the project and then select properties. Then we need to select libraries in Java Build Path and then select Add External Libraries. In this tab select File System > usr > lib> Hadoop. Select all the jar files available in Hadoop folder and OK. We should also add the jar files (libraries) available in client folder.

A screenshot of a cell phone

Description automatically generated

A screenshot of a computer

Description automatically generated

The next step is to create a new class. For this we will right click on src and then new > class. Then a new class will be created in which we can enter our code, check for errors and execute it to get a desired output.

A screenshot of a computer

Description automatically generated

After executing our code without any errors, we need to export our project as a JAR file. So, for this we should right click on project, select Export. Then select Java > JAR file > Project name and then Enter the destination address where you want to save the JAR file and click finish.

A screenshot of a social media post

Description automatically generated

1. Counting the frequency of words in the given input with MapReduce algorithm.

**Code:**

Here are we are defining three classes: Main Class, Reducer Class and the Mapper Class.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**Output:**

Here we are using the command Hadoop jar WordCount.jar WordCount input/sample.txt output. This command reads the jar file, considers sample.txt as the input file and implements the functionality in the class WordCount which is present in the JAR on the input file and creates another file with the name output and stores the corresponding output in that file.

In order to see the output of the Map Reduce program we will use the command Hadoop fs -cat output/part-r-00000.

A screenshot of a social media post

Description automatically generated

A screenshot of a social media post

Description automatically generated

A screenshot of a social media post

Description automatically generated

Below is the screenshot of the output of the above program in Hadoop User Interface.

A screenshot of a social media post

Description automatically generated

1. Counting the frequency of words in given text file that starts with letter ‘a’.

**Code:**

Here in order to find the frequency of words in the file which starts with a we are using an if

Statement for filtering out the words which start with ‘a’ in the reducer class.

A screenshot of a computer

Description automatically generated

**Output:**

This is the screenshot of the output in the terminal which is obtained by using the command -cat.

A screenshot of a social media post

Description automatically generated

This is the screenshot of the output in Hadoop User Interface.

A screenshot of a social media post

Description automatically generated

1. Determine the prime number in input.

**Code:**

This is the screenshot of the code of reducer class where the functionality for determining if a number is a prime number or not is written.

A screenshot of a computer

Description automatically generated

**Output:**

This is the screenshot of the output in Hue.

A screenshot of a social media post

Description automatically generated