**EXPERIMENT 2**

| Objective :  Theory:  Performance: | Write a program to implement a word count program using MapReduce  Steps for the Word Count Program With MapReduce and Java:  In Hadoop, MapReduce is a computation that decomposes large manipulation jobs into individual tasks  that can be executed in parallel across a cluster of servers. The results of tasks can be joined together to  compute final results.  MapReduce consists of 2 steps:  **Map Function –** It takes a set of data and converts it into another set of data, where individual elements  are broken down into tuples (Key-Value pair).  **Example –** (Map function in Word Count)    **Reduce Function –** Takes the output from Map as an input and combines those data tuples into a  smaller set of tuples.  **Example –** (Reduce function in Word Count)    **Work Flow of the Program**  **1. Splitting –** The splitting parameter can be anything, e.g. splitting by space, comma, semicolon, or even by a new line (‘\n’).  **2. Mapping –** as explained above.  **3. Intermediate splitting –** the entire process in parallel on different clusters. In order to group them in “Reduce Phase” the similar KEY data should be on the same cluster.  **4. Reduce –** it is nothing but mostly group by phase.  **5. Combining –** The last phase where all the data (individual result set from each cluster) is combined together to form a result    Download[WordCount.java](https://drive.google.com/file/d/1kSkslwNJZ30K3h8d5vwydkmrVOVGsVNi/view?usp=sharing) program file before you start.  **Steps to run WordCount Program on Hadoop:**   1. Make sure Hadoop and Java are installed properly   **hadoop version**  **javac -version**   1. Create a directory on the Desktop named Lab and inside it create two folders; one called “Input” and the other called “tutorial\_classes”.   [You can do this step using GUI normally or through terminal commands]  **cd Desktop**  **mkdir Lab**  **mkdir Lab/Input**  **mkdir Lab/tutorial\_classes**   1. Add the file attached with this document “WordCount.java” in the directory Lab 2. Add the file attached with this document “input.txt” in the directory Lab/Input. 3. Type the following command to export the hadoop classpath into bash.   **export HADOOP\_CLASSPATH=$(hadoop classpath)**  Make sure it is now exported.  **echo $HADOOP\_CLASSPATH**   1. It is time to create these directories on HDFS rather than locally. Type the following commands.   **hadoop fs -mkdir /WordCountTutorial**  **hadoop fs -mkdir /WordCountTutorial/Input**  **hadoop fs -put Lab/Input/input.txt /WordCountTutorial/Input**   1. Go to localhost:9870 from the browser, Open “Utilities → Browse File System” and you should see the directories and files we placed in the file system. 2. Then, back to local machine where we will compile the WordCount.java file. Assuming we are currently in the Desktop directory.   **cd Lab**  **javac -classpath $HADOOP\_CLASSPATH -d tutorial\_classes WordCount.java**  Put the output files in one jar file (There is a dot at the end)  **jar -cvf WordCount.jar -C tutorial\_classes .**   1. Now, we run the jar file on Hadoop.   **hadoop jar WordCount.jar WordCount /WordCountTutorial/Input /WordCountTutorial/Output**   1. Output the result:   **hadoop dfs -cat /WordCountTutorial/Output/\*** |
| --- | --- |
| **Deliverables:**  **Summary:** | Step wise Screen shots of the word count program execution  Write summary in your own words with the achieved result |

Wo