**BIG DATA ANALYTICS**

**LAB4**

**AIM: Write a map-reduce program to count the frequencies of words from a distributed storage source and understand the phases involved in map-reduce programming.**

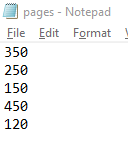
**EXERCISE:**

**Step 1.1: Create a file named 'pages.txt' in the local file system. Store line by line content as shown below. Each line data represents the number of pages of a sample book.**

**350 250 150 450 120**

**Solution:**

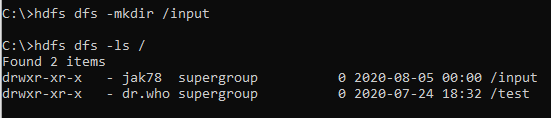
* Number of pages is stored line by line.

****

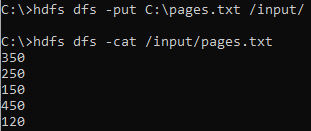
**Step 1.2: Put the file from the local file system to hdfs with a folder named 'input'. Confirm the presence of above data.**

**Solution:**

* Using mkdir we will create a folder named ‘input’ and using ls command we can see that the input folder is created successfully.

****

* Using the **put** command we can put the file in the ‘input’ folder to hdfs.
* We can verify the data of the text file using the **cat** command.

****

**Step 1.3: Write a map and reduce functions to split the books into the following two categories: (a) Big Books (b) Small Books**

**Books which have more than 300 pages should be in the big book category. Books which have less than 300 pages should be in the small book category. Count the number of books in each category. Store the output as follows as result file within hdfs 'output’ folder.**

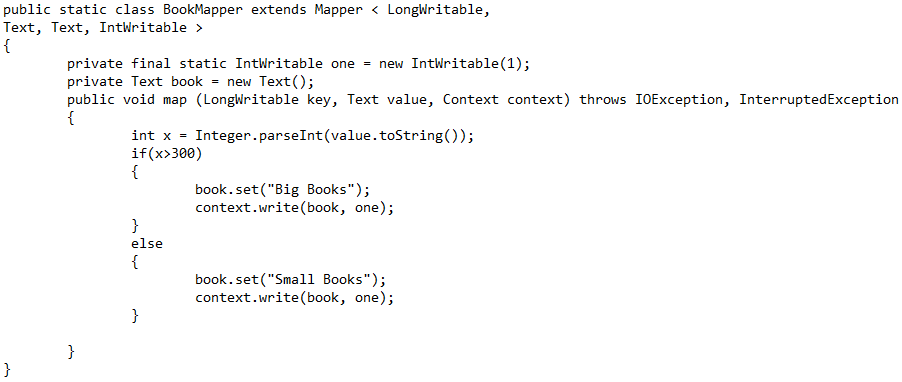
**Book Category Count of the books**

**"Big Books" 2**

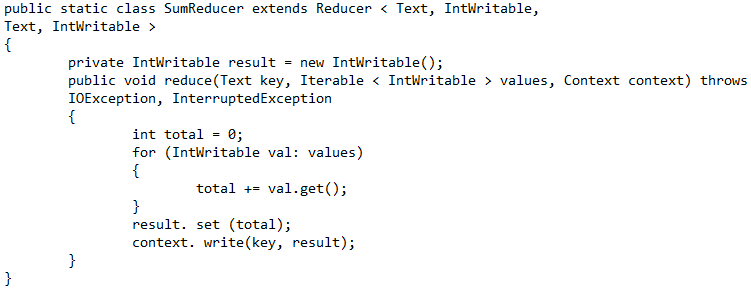
**"Small Books" 3**

**Solution:**

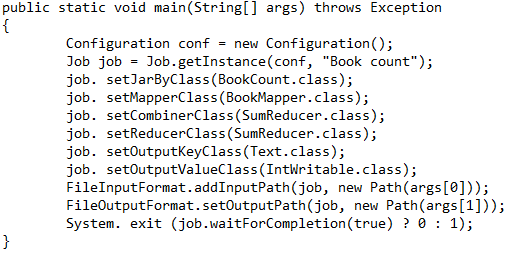
* **Map function**

****

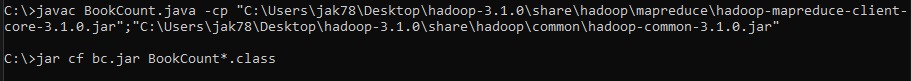
* **Reduce function**

****

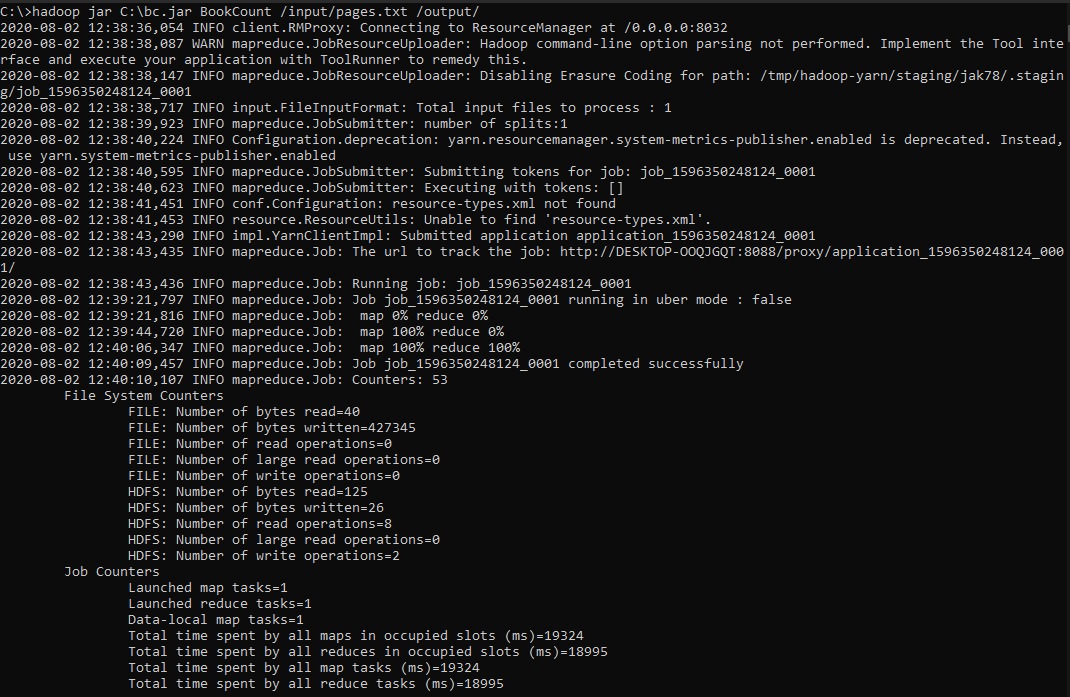
* **Driver function**

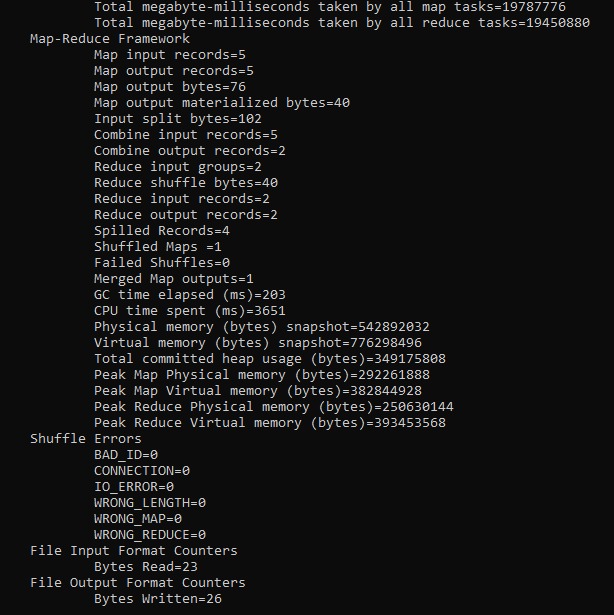
****

* Compile the program using the following command.

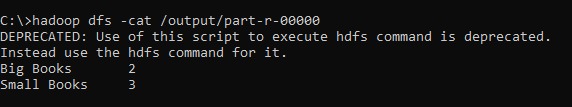
****

* Store the output within the hdfs ‘output’ folder.

****

****

Output:

****