Computer Science & Information Systems

**Big Data Systems – Hadoop Lab Sheet 3**

**Intro to Hadoop MapReduce**

1. Objective:

Students should be able to

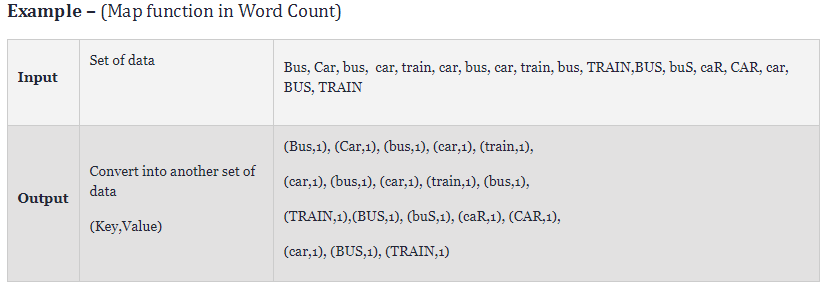
* 1. Understand the use cases related to MapReduce programming
  2. Write simple MapReduce Word Count programme
  3. Execute the MapReduce code on Hadoop cluster
  4. Observe the outcomes of MapReduce codes

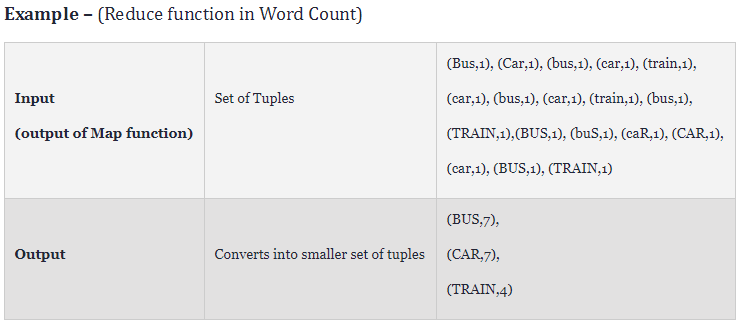
In Hadoop, MapReduce is a computation that decomposes large manipulation jobs into individual tasks that can be executed in parallel across a cluster of servers. The results of tasks can be joined together to compute final results.

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).

**Reduce Function** – Takes the output from Map as an input and combines those data tuples into a smaller set of tuples.

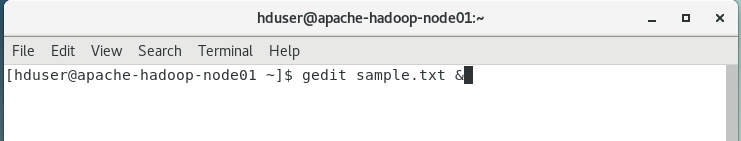


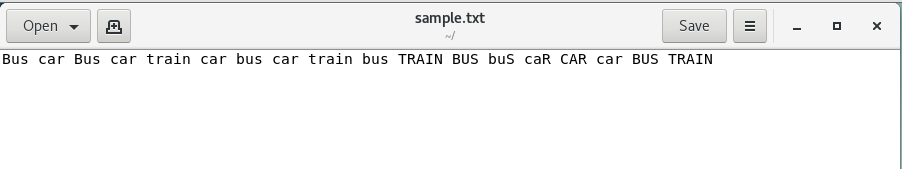


**Pre-requisites:**

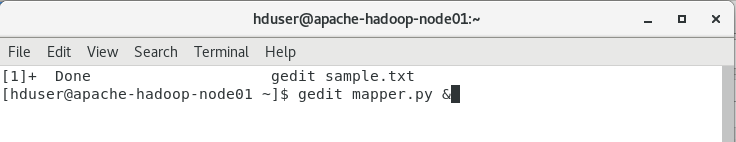
1. Hadoop should be installed
2. Hadoop cluster should be up and running
3. Python should be installed
4. Steps to be performed:
5. Create a file with the following content and name it sample.txt.

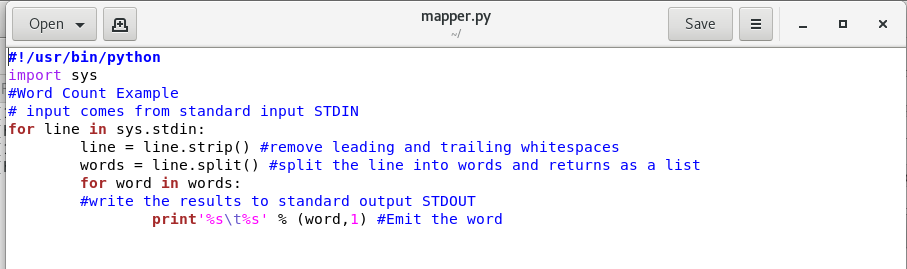
Bus car Bus car train car bus car train bus TRAIN BUS buS caR CAR car BUS TRAIN





1. Write the mapper code in mapper.py





Mapper code - mapper.py

*#!/usr/bin/python*

*import sys*

*#Word Count Example*

*# input comes from standard input STDIN*

*for line in sys.stdin:*

*line = line.strip() #remove leading and trailing whitespaces*

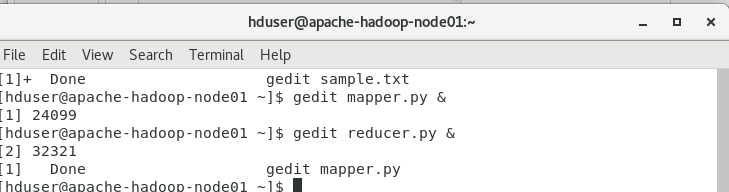
*words = line.split() #split the line into words and returns as a list*

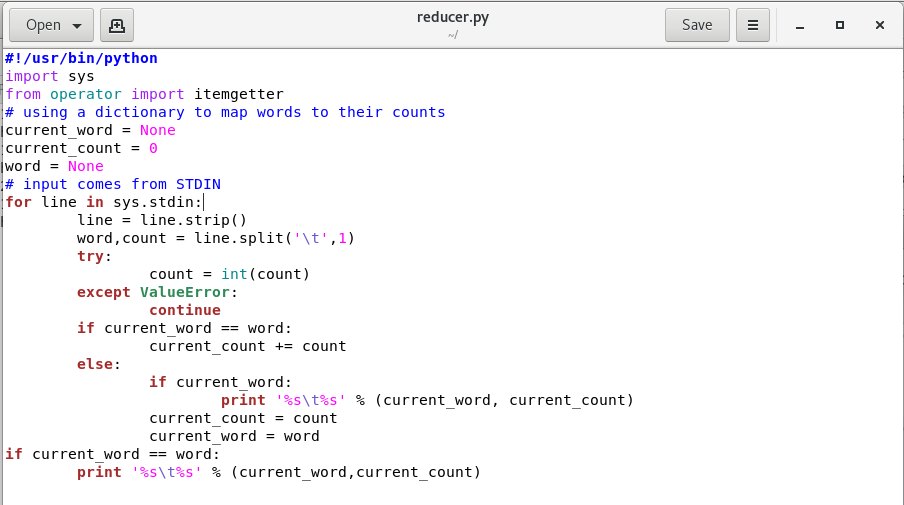
*for word in words:*

*#write the results to standard output STDOUT*

*print'%s\t%s' % (word,1) #Emit the word*

1. Write the reducer code in reducer.py





Reduce code – reducer.py

*#!/usr/bin/python*

*import sys*

*from operator import itemgetter*

*# using a dictionary to map words to their counts*

*current\_word = None*

*current\_count = 0*

*word = None*

*# input comes from STDIN*

*for line in sys.stdin:*

*line = line.strip()*

*word,count = line.split('\t',1)*

*try:*

*count = int(count)*

*except ValueError:*

*continue*

*if current\_word == word:*

*current\_count += count*

*else:*

*if current\_word:*

*print '%s\t%s' % (current\_word, current\_count)*

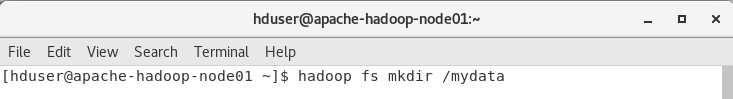
*current\_count = count*

*current\_word = word*

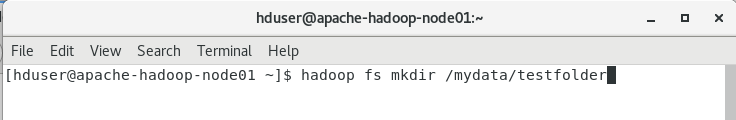
*if current\_word == word:*

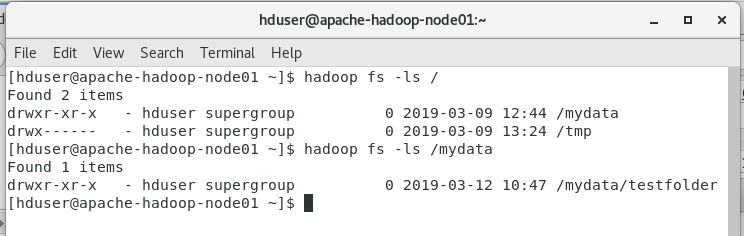
*print '%s\t%s' % (current\_word,current\_count)*

1. Create directory in Hadoop file system where you want to store your data files
   1. hadoop fs -mkdir /mydata

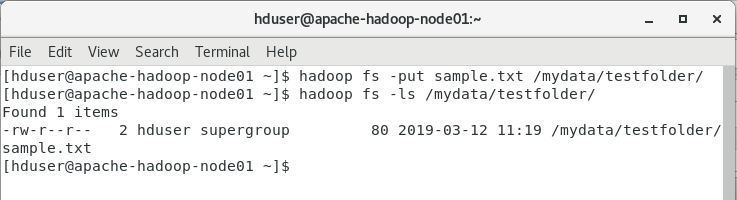


* 1. hadoop fs -mkdir /mydata/testfolder



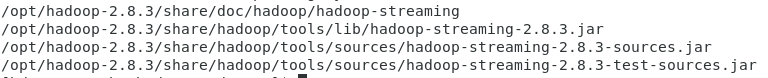


1. Move the data file i.e. sample.txt to Hadoop file system
   1. hadoop fs -put sample.txt /mydata/testfolder/

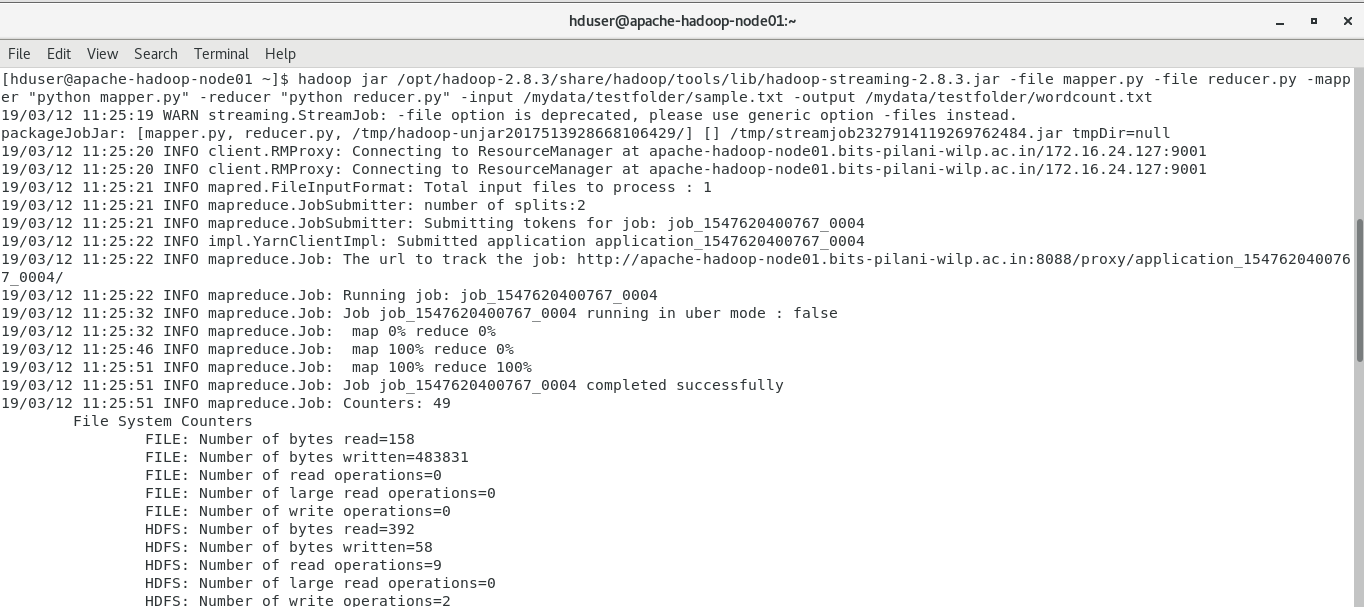


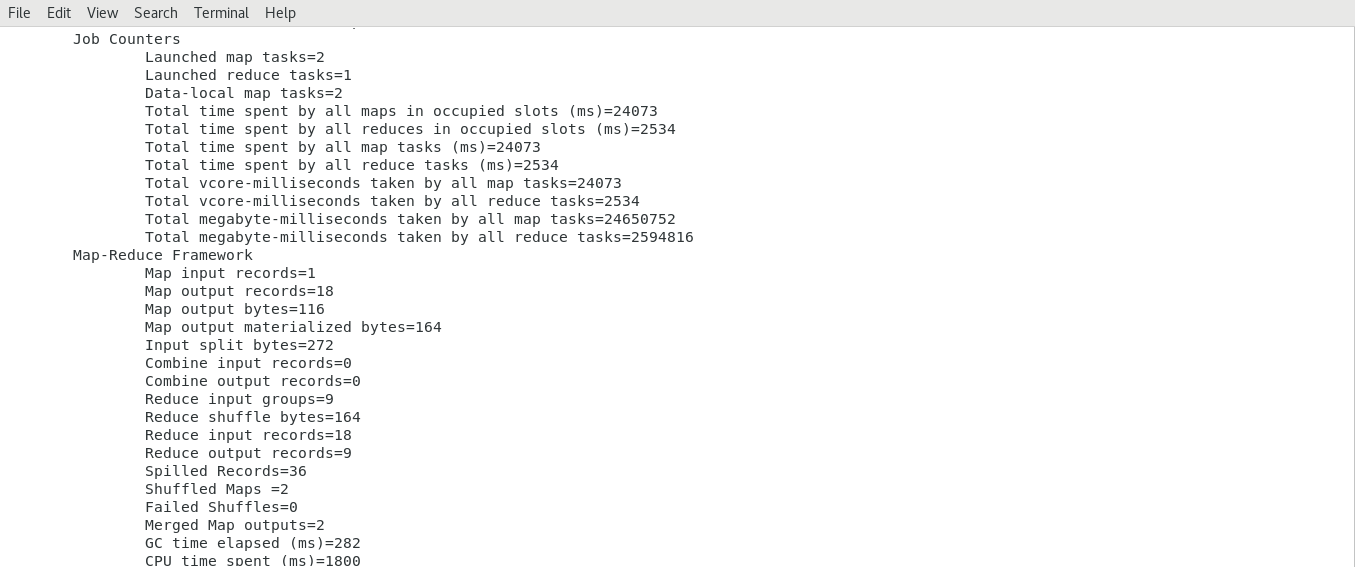
1. Find Hadoop streaming location
   1. find / -name Hadoop\*stream\*

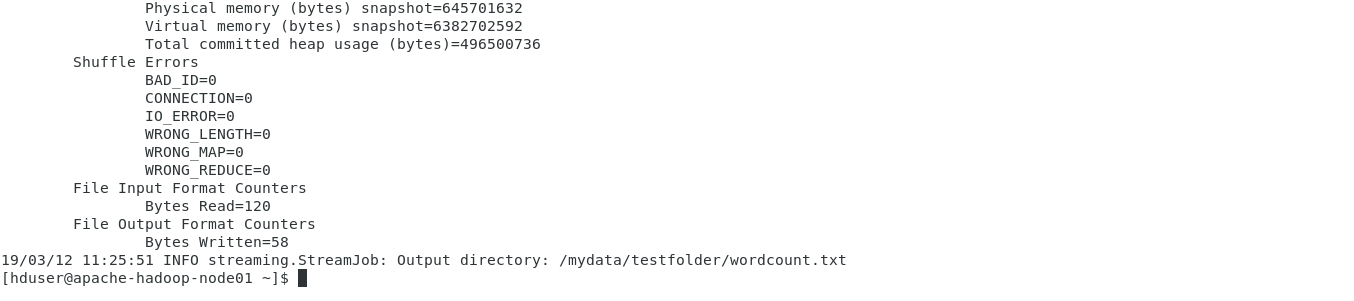




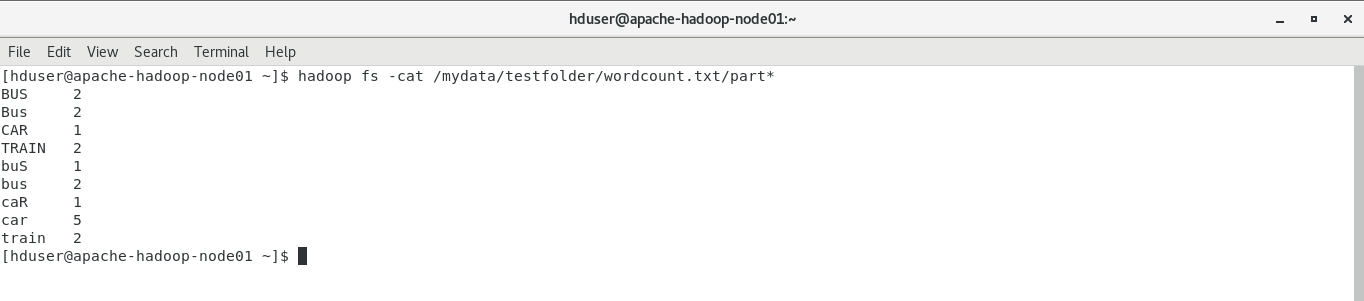
1. Run map reduce using streaming
   1. hadoop jar /opt/hadoop-2.8.3/share/hadoop/tools/lib/hadoop-streaming-2.8.3.jar -file mapper.py -file reducer.py -mapper "python mapper.py" -reducer "python reducer.py" -input /mydata/testfolder/sample.txt -output /mydata/testfolder/wordcount.txt







1. Check the output
   1. hadoop fs -cat /mydata/testfolder/wordcount.txt/part\*



1. Check the Hadoop application information.
2. Outputs/Results:

Students should be able to use Hadoop setup for

* Writing the simple MapReduce Word Count programme
* Executing the MapReduce code on Hadoop cluster
* Observing the outcomes of MapReduce codes

1. Observations:

Students carefully show observe for

* The use case for MapReduce programming paradigm
* Ways to convert the traditional code into MapReduce program
* Ways to execute MapReduce program

1. References:
2. [Example: WordCount v2.0](https://hadoop.apache.org/docs/current/hadoop-mapreduce-client/hadoop-mapreduce-client-core/MapReduceTutorial.html#Example:_WordCount_v2.0)
3. [Hadoop Streaming](https://hadoop.apache.org/docs/r1.2.1/streaming.html)