**Hadoop Streaming Mapreduce using Python**

Hadoop Streaming is a utility that comes with the Hadoop distribution. It can be used to execute programs for big data analysis. Hadoop streaming can be performed using languages like Python, Java, PHP, Scala, Perl, UNIX, and many more. The utility allows us to create and run Map/Reduce jobs with any executable or script as the mapper and/or the reducer.

This is wordcount project using MapReduce streaming.

1. Create project folder “mapreduce” using **sudo mkdir mapreduce**
2. Change directory **cd mapreduce.**
3. Create mapper.py project file. **sudo nano mapper.py**
4. Write program **and save program file. Ctrl + O, Press Enter and Ctrl + Y**

#!/usr/bin/python3

"""mapper.py"""

import sys

# input comes from standard input

for line in sys.stdin:

# remove leading and trailing whitespace

line = line.strip()

# split the line into words

words = line.split()

# increase counters

for word in words:

# write the results to strdout(standard output)

# tab-delimited; the trivial word count is 1

# Output of Mapper Job is input to Reducer <--

print('%s\t%s' % (word, 1))

1. Make sure you make python file executable code. **sudo chmod 777 mapper.py** or sudo **chmod +x mapper.py**
2. Create mapper.py project file. **sudo nano reducer.py**
3. Write program **and save program file. Ctrl + O, Press Enter and Ctrl + Y**

#!/usr/bin/python3

"""reducer.py"""

import sys

current\_word = None

current\_count = 0

word = None

# input comes from STDIN

for line in sys.stdin:

line = line.strip()

# parse the input we got from mapper.py

# word & count are seperated by <tab> delimited

# 1 -> No of splits

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

# convert count (currently a string) to integer data

try:

count = int(count)

except ValueError:

# count was not a number, so silently

# ignore/discard this line

continue

# this IF-switch only works because Hadoop sorts map output

# by key (here key is word) before it is passed to the reducer

if current\_word == word:

current\_count += count

else:

if current\_word:

# write result to stdoutput seperated by tab

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

# making current\_word = word

current\_count = count

current\_word = word

# do not forget to output the last word if needed!

if current\_word == word:

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

1. Make sure the file has execution permission **sudo chmod +x reducer.py**
2. echo "example mapreduce wordcount code easy code example" | /home/hadoop/mapreduce/mapper.py
3. echo "example mapreduce wordcount code easy code example" | /home/hadoop/mapreduce/mapper.py | sort -k1,1 | /home/hadoop/mapreduce/reducer.py
4. Create a file with the some content and name it wordfile.txt using **sudo nano wordfile.txt**
5. cat wordfile.txt | python3 mapper.py
6. cat wordfile.txt | python3 mapper.py | sort -k1,1 | python3 reducer.py
7. hdfs dfs -mkdir /wordcount
8. hdfs dfs -put /home/hadoop/mapreduce/wordfile.txt /wordcount/
9. Run Mapreduce Streaming Job
10. hadoop jar /usr/local/hadoop/share/hadoop/tools/lib/hadoop-streaming-2.7.6.jar -file /home/hadoop/mapreduce/mapper.py -mapper mapper.py -file /home/hadoop/mapreduce/reducer.py -reducer reducer.py -input /wordcount -output /wordcount/output/
11. hadoop fs -cat /wordcount/output/part-00000