1. *Implementation of MapReduce program for Wordcount using python.*

*#!/usr/bin/python*

**import** **sys**

*#--- get all lines from stdin ---*

**for** line **in** sys.stdin:

*#--- remove leading and trailing whitespace---*

line = line.strip()

*#--- split the line into words ---*

words = line.split()

*#--- output tuples [word, 1] in tab-delimited format---*

**for** word **in** words:

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

*#!/usr/bin/python*

**import** **sys**

*# maps words to their counts*

word2count = {}

*# input comes from STDIN*

**for** line **in** sys.stdin:

*# remove leading and trailing whitespace*

line = line.strip()

*# parse the input we got from mapper.py*

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

*# convert count (currently a string) to int*

**try**:

count = int(count)

**except** **ValueError**:

**continue**

**try**:

word2count[word] = word2count[word]+count

**except**:

word2count[word] = count

*# write the tuples to stdout*

*# Note: they are unsorted*

**for** word **in** word2count.keys():

**print** '**%s\t%s**'% ( word, word2count[word] )

**1. Giving permission to mapper and reducer files in local file system**

chmod a+x /home/cloudera/Desktop/WordCount/mapper.py

chmod a+x /home/cloudera/Desktop/WordCount/reducer.py

**2. Execution of Word Count with Combiner**

hadoop jar /usr/lib/hadoop-0.20-mapreduce/contrib/streaming/hadoop-streaming-2.6.0-mr1-cdh5.8.0.jar -input /WordCount\_INP -output /WordCount\_OUT\_WC -mapper /home/cloudera/Desktop/WordCount/mapper.py -combiner /home/cloudera/Desktop/WordCount/reducer.py -reducer /home/cloudera/Desktop/WordCount/reducer.py

**3. Execution of Word Count without Combiner**

hadoop jar /usr/lib/hadoop-0.20-mapreduce/contrib/streaming/hadoop-streaming-2.6.0-mr1-cdh5.8.0.jar -input /WordCount\_INP -output /WordCount\_OUT\_NC -mapper /home/cloudera/Desktop/WordCount/mapper.py -reducer /home/cloudera/Desktop/WordCount/reducer.py

1. ***Map function for maximum temperature in Python***

#!/usr/bin/env python

import re

import sys

for line in sys.stdin:

val = line.strip()

(year, temp, q) = (val[15:19], val[87:92], val[92:93])

if (temp != "+9999" and re.match("[01459]", q)):

print "%s\t%s" % (year, temp)

Reduce function

#!/usr/bin/env python

import sys

(last\_key, max\_val) = (None, 0)

for line in sys.stdin:

(key, val) = line.strip().split("\t")

if last\_key and last\_key != key:

print "%s\t%s" % (last\_key, max\_val)

(last\_key, max\_val) = (key, int(val))

else:

(last\_key, max\_val) = (key, max(max\_val, int(val)))

if last\_key:

print "%s\t%s" % (last\_key, max\_val)

**3. Write a MapReduce program to find Dept wise salary.**

**Empno EmpName Dept Salary**

**Mapper.py**

#!/usr/bin/env python

import sys

for line in sys.stdin:

line=line.strip()

words=line.split()

size=len(words)

key=word[size-1]+”-”+word[size-2]+”’-”+word[size-6];

print '%s\t%s' %(words[size-2],words[size-1])

**Reducer.py**

#!/usr/bin/env python

import sys

current\_dept=None

dept=None

current\_sal=0

for line in sys.stdin:

line=line.strip()

dept,sal=line.split('\t',1)

try:

sal=int(sal)

except ValueError:

continue

if current\_dept==dept:

current\_sal+=sal

else:

if current\_dept:

print '%s\t%s' %(current\_dept,current\_sal)

current\_dept=dept

current\_sal=sal

if current\_dept==dept:

print '%s\t%s' %(current\_dept,current\_sal)

**Input.txt**

1011 Abc CSE 50000

1012 Def ECE 45000

1013 Efg Mech 45000

1014 Ghi CSE 55000

1015 Jkl CSE 75000

1016 Mno Mech 35000

1017 Pqr ECE 46000

1018 Stu EEE 25000

1019 Vwx CSE 31000

1020 Yzz EEE 25000

**Output**

[cloudera@quickstart ~]$ hadoop fs -ls /4094\_out\_Dept

Found 2 items

-rw-r--r-- 1 cloudera supergroup 0 2018-09-26 01:55 /4094\_out\_Dept/\_SUCCESS

-rw-r--r-- 1 cloudera supergroup 42 2018-09-26 01:55 /4094\_out\_Dept/part-00000

[cloudera@quickstart ~]$ hadoop fs -cat /4094\_out\_Dept/part-00000

**CSE 211000**

**ECE 91000**

**EEE 50000**

**Mech 80000**

[cloudera@quickstart ~]$