MapReduce Implementation

Dataset:

The dataset is a household power consumption dataset taken from kaggle. This dataset contains the following fields:

- Date
- Time
- Global Active Power
- Global Reactive Power
- Voltage
- Global Intensity
- Sub Metering-1
- Sub Metering-2
- Sub Metering-3

This dataset contains power details for every date and for every minute from 1st January 2008 to 31st December 2010. Hence there are 1578240 records. A small snapshot of the dataset is shown below.

```
Date;Time;Global_active_power;Global_reactive_powe
r;Voltage;Global_intensity;Sub_metering_1;Sub_mete
ring_2;Sub_metering_3
1/1/2008;00:00:00;1.620;0.070;241.250;6.600;0.000;
0.000;18.000
```

```
1/1/2008;00:01:00;1.626;0.072;241.740;6.600;0.000;
0.000;18.000
1/1/2008;00:02:00;1.622;0.072;241.520;6.600;0.000;
0.000;18.000
1/1/2008;00:03:00;1.612;0.070;240.820;6.600;0.000;
0.000;18.000
1/1/2008;00:04:00;1.612;0.070;240.800;6.600;0.000;
0.000;18.000
1/1/2008;00:05:00;1.546;0.000;240.660;6.400;0.000;
0.000;17.000
```

Problem Statement:

The problem statement is to find the ***maximum voltage for each day*** of each of the years. Processing 1578240 records will take a lot of time. Hence Map Reduce will be best suited for this problem.

Mapper code:

```
import sys

for line in sys.stdin:
   line = line.strip()
   splits = line.split(";")
   if(splits[0]!='Date' and splits[4]!='Voltage'):
        print(splits[0]+" "+splits[4])
```

The preprocessing and mapping is done in the mapper code. Only the date and voltage values are returned by mapper.

Reducer code:

```
import sys

dictionary={}
for line in sys.stdin:
    line = line.strip()
    date, voltage = line.split(" ")
    dictionary.setdefault(date,[])
    dictionary[date].append(float(voltage))

for i in dictionary:
    lists = list(dictionary[i])
    print(i+" "+str(max(lists)))
```

The shuffling, merging and reducing phase is done in reducer code. The output of this reducer code is the list of dates and maximum voltage of that dates.

Driver code:

```
hadoop jar /usr/local/Cellar/hadoop/3.3.0/libexec/
share/hadoop/tools/lib/hadoop-streaming-3.3.0.jar
-file /Users/karanrajmokan/Desktop/MapReduce/
mapper.py -mapper mapper.py -file /Users/
karanrajmokan/Desktop/MapReduce/reducer.py
-reducer reducer.py -input /user/Hadoop/MapReduce/
household_power_consumption.txt -output /user/
Hadoop/MapReduce/maxVoltages
```

The above code is used to run the Hadoop map and reduce processes.

- The Hadoop streaming jar is used to start and run map and reduce processes.
- There are two files (one for mapper, other for reducer) which are mapped to their corresponding local system location.
- The input file (i.e dataset) is already put in the Hadoop directory /user/Hadoop/ MapReduce/ and it will be directly taken from there.
- The output directory should be explicitly mentioned (i.e /user/Hadoop/MapReduce/ maxVoltages) and it will be stored in that directory after the implementation.

Output:

As expected, the total number of records should be 3*365 + 1(leap day for the year 2008) = 1096. The first 10 output records are shown below.

```
1/1/2008 246.65
1/1/2009 249.55
1/1/2010 248.81
1/10/2008 247.7
1/10/2009 247.72
1/10/2010 246.58
1/11/2008 246.65
1/11/2009 247.65
1/11/2010 253.08
1/12/2008 250.92
```

Steps to install Hadoop:

1. Installing hadoop using homebrew

```
brew install hadoop
```

2. Check whether it's installed

```
hadoop version
```

3. Locate the working directory

```
cd /usr/local/Cellar/hadoop/3.3.0/libexec/sbin/
```

4. To create directories in hadoop

```
    hadoop fs -mkdir /user
    hadoop fs -mkdir /user/Hadoop
    hadoop fs -mkdir /user/Hadoop/MapReduce
```

5. To put files in the hadoop directory

```
hdfs dfs -put /localdir/power_consumption.txt
/user/Hadoop/MapReduce/
```

6. Copy output file to local machine

```
hdfs dfs -get /user/Hadoop/MapReduce/
maxVoltages/part-00000 /localdir/
```

Team Members:

1. Karanraj M (17PW18)
2. Sathwik Ch (17PW32)