CODE OUTPUT

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer screen

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

MAPPER.py

#!/usr/bin/env python3

import json

import sys

# Read the JSON file line by line from stdin

for line in sys.stdin:

try:

if not line:

continue

line = line.strip().strip(',')

# Convert the line (which is a string representing a JSON object) to a dictionary

json\_dict = json.loads(line)

name = json\_dict['name']

runs = json\_dict['runs']

balls = json\_dict['balls']

local\_strike\_rate = (runs / balls) \* 100

print("%s\t%s" % (name, local\_strike\_rate))

except Exception as e:

sys.stderr.write("Error processing line: %s\n" % e)

REDUCER.py

#!/usr/bin/env python3

import sys

current\_name = None

total\_strike\_rate = 0

total\_matches = 0

# Read input from standard input

for line in sys.stdin:

try:

# Parse input

name, local\_strike\_rate = line.strip().split('\t')

local\_strike\_rate = float(local\_strike\_rate)

# Update total strike rate and total matches

if current\_name == name:

total\_strike\_rate += local\_strike\_rate

total\_matches += 1

else:

if current\_name:

# Output average strike rate for the previous name

average\_strike\_rate = total\_strike\_rate / total\_matches

print('%s\t%.3f' % (current\_name, average\_strike\_rate))

# Update current name and reset counters

current\_name = name

total\_strike\_rate = local\_strike\_rate

total\_matches = 1

except Exception as e:

# Print error message to stderr

sys.stderr.write("Error processing line: {}\n".format(line.strip()))

sys.stderr.write(str(e) + "\n")

# Output average strike rate for the last name

if current\_name:

average\_strike\_rate = total\_strike\_rate / total\_matches

print('%s\t%.3f' % (current\_name, average\_strike\_rate))

CODE:

SETTING YOUR CLIENT AND SERVER FOR MOUNTING LARGE DATA FILES ON UBUNTU WSL

sudo add-apt-repository ppa:n-muench/programs-ppa

sudo apt-get update

sudo apt-get install filezilla

filezilla

On Windows download the latest version of FileZilla

Enter your IP address in host name, username and password

cd ~

mkdir StrikeRate

cd StrikeRate

touch mapper.py

touch reducer.py

touch Strike\_input\_data.json

nano mapper.py

nano reducer.py

nano Strike\_input\_data.json

chmod +x mapper.py reducer.py

cd ~/hadoop/hadoop-3.3.6

sbin/start-dfs.sh

sbin/start-yarn.sh

export HADOOP\_HOME=~/hadoop/hadoop-3.3.6

cd ~

hdfs dfs -ls

hdfs dfs -mkdir ./StrikeRateData

hdfs dfs -ls

hdfs dfs -put Strike\_input\_data.json ./StrikeRateData

ls ~/hadoop/hadoop-3.3.6/share/hadoop/tools/lib/

hadoop jar ~/hadoop/hadoop-3.3.6/share/hadoop/tools/lib/hadoop-streaming-3.3.6.jar -input ./StrikeRateData/Strike\_input\_data.json -output ./StrikeRateData/Strike\_rate\_output.txt -mapper ./mapper.py -file mapper.py -reducer ./reducer.py -file reducer.py

hdfs dfs -ls ./StrikeRateData/

hdfs dfs -cat ./StrikeRateData/Strike\_rate\_output.txt/\*