# Hands-on Assignment 2

## Task 1

7.show the directory with its new owner

```
eric@eric-VirtualBox:~$ sudo chown -R eric:eric /usr/local/hadoop
eric@eric-VirtualBox:~$ ls -ld /usr/local/hadoop
drwxr-xr-x 9 eric eric 4096 Sep 14 09:39 /usr/local/hadoop
```

8.the directory for java

```
eric@eric-VirtualBox:~$ readlink -f /usr/bin/java | sed "s:/bin/java::"
/usr/lib/jvm/java-8-openjdk-amd64/jre
```

10. close .bashrc file and execute command: “. .bashrc”

```
eric@eric-VirtualBox:~$ . .bashrc
eric@eric-VirtualBox:~$ echo $JAVA_HOME
/usr/lib/jvm/java-8-openjdk-amd64/jre
```

12.show the directory with its new owner

```
eric@eric-VirtualBox:~$ sudo chown -R eric:eric /app/hadoop/tmp
eric@eric-VirtualBox:~$ ls -ld /app/hadoop/tmp
drwxr-xr-x 2 eric eric 4096 Oct 28 18:08 /app/hadoop/tmp
```

13.show the output of the ‘core-site.xml’ file

```
eric@eric-VirtualBox: ~
eric@eric-VirtualBox:~$ cat /usr/local/hadoop/etc/hadoop/core-site.xml
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

<configuration>
<property>
<name>hadoop.tmp.dir</name>
<value>/app/hadoop/tmp</value>
</property>
<property>
<name>fs.default.name</name>
<value>hdfs://localhost:9000</value>
</property>
</configuration>
eric@eric-VirtualBox:~$
```

14. show the output of the ‘hdfs-site.xml’ file

```
eric@eric-VirtualBox:~$ cat /usr/local/hadoop/etc/hadoop/hdfs-site.xml
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

<configuration>
<property>
<name>dfs.replication</name>
<value>1</value>
</property>
</configuration>
eric@eric-VirtualBox:~$
```

15. show the output of the ‘mapred-site.xml’ file

```
eric@eric-VirtualBox:~$ cat /usr/local/hadoop/etc/hadoop/yarn-site.xml
<?xml version="1.0"?>
<!--
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->
<configuration>

<!-- Site specific YARN configuration properties -->
<property>
<name>yarn.nodemanager.aux-services</name>
<value>mapreduce_shuffle</value>
</property>
</configuration>
```

16. show the output of the ‘yarn-site.xml’ file

```
eric@eric-VirtualBox:~$ cat /usr/local/hadoop/etc/hadoop/mapred-site.xml
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

<configuration>
<property>
<name>mapreduce.framework.name</name>
<value>yarn</value>
</property>
</configuration>
```

20.check services

```
eric@eric-VirtualBox:~$ jps
3265 ResourceManager
2916 DataNode
3125 SecondaryNameNode
2728 NameNode
3721 Jps
3418 NodeManager
```

Commands:

2.examine files in HDFS cluster

```
eric@eric-VirtualBox:~$ cd
eric@eric-VirtualBox:~$ hdfs dfs -ls /
eric@eric-VirtualBox:~$
```

3.create a directory in HDFS cluster

```
eric@eric-VirtualBox:~$ hdfs dfs -mkdir /user
eric@eric-VirtualBox:~$ hdfs dfs -ls /
Found 1 items
drwxr-xr-x   - eric supergroup          0 2020-10-31 21:59 /user
```

4.examine files again

```
eric@eric-VirtualBox:~$ hdfs dfs -mkdir /user
eric@eric-VirtualBox:~$ hdfs dfs -ls /
Found 1 items
drwxr-xr-x   - eric supergroup          0 2020-10-31 21:59 /user
```

5.create a new directory

```
eric@eric-VirtualBox:~$ hdfs dfs -mkdir /user/eric
eric@eric-VirtualBox:~$ hdfs dfs -ls /user/
Found 1 items
drwxr-xr-x   - eric supergroup          0 2020-10-31 22:01 /user/eric
```

6.examine the new directory

```
eric@eric-VirtualBox:~$ hdfs dfs -mkdir /user/eric
eric@eric-VirtualBox:~$ hdfs dfs -ls /user/
Found 1 items
drwxr-xr-x   - eric supergroup          0 2020-10-31 22:01 /user/eric
```

8.upload the file in the cluster

```
eric@eric-VirtualBox:~$ touch demo.txt
eric@eric-VirtualBox:~$ gedit demot.txt
eric@eric-VirtualBox:~$ hdfs dfs -copyFromLocal demo.txt /user/eric/
```

10.see the content of the file

```
eric@eric-VirtualBox:~$ hdfs dfs -cat /user/eric/demo.txt
It's my first time to use hadoop
```

## Task2

## A

**mapper_Wei_A.py**

```
#!/usr/bin/python3

import string

import sys



# get all lines from stdin

for line in sys.stdin:

    val = line.strip()
```

```
    words = val.split(' ')
    for word in words:
        for letter in word:
            # remove punctuations from the word
            punc = '''!()-[]{};:'"\\, <>./?@#$%^&*_~'''
            if letter in punc:
                word = word.replace(letter, '')
        # uniform the word in lower case
        word = word.lower()
        print ('%s\t%s' % (word, 1))
```

**reducer_Wei_A.py**

```
#!/usr/bin/python3
import sys
from operator import itemgetter

items = []
new_items = []
(last_key, count) = (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, count))
        items.append((last_key, count))
        (last_key, count) = (key, 1)
    else:
```

```
        (last_key, count) = (key, count + 1)
if last_key:
    # print('%s\t%s' % (last_key, count))
    items.append((last_key, count))

items = sorted(items, key = itemgetter(1), reverse = True)
for item in items:
            new_items.append('%s\t%s' % (item[0], item[1]))

for newItem in new_items[0:10]:
            print(newItem)
```

**result:**

```
eric@eric-VirtualBox:~$ hdfs dfs -ls /user/eric/outputA
Found 2 items
-rw-r--r--   1 eric supergroup          0 2020-11-15 14:00 /user/eric/outputA/_SUC
CESS
-rw-r--r--   1 eric supergroup         61 2020-11-15 14:00 /user/eric/outputA/part
-00000
eric@eric-VirtualBox:~$ hdfs dfs -cat /user/eric/outputA/part-00000
the     41
of      30
and     17
is      12
to      10
a       9
science 8
be      6
in      6
it      6
```

## B

**mapper_Wei_B.py**

```
#!/usr/bin/python3
import string
```

```
import sys

for line in sys.stdin:
    line = line.strip()
    words = line.split('   ')
    for word in words[2:5:2]:
        print(str(word), end ='\t')
    print()
```

**reducer_Wei_B.py**

```
#!/usr/bin/python3
import sys
from math import sqrt


def update(existingAggregate, newValue):
    (count, mean, M2) = existingAggregate
    count += 1
    delta = newValue - mean
    mean += delta / count
    delta2 = newValue - mean
    M2 += delta * delta2
    return (count, mean, M2)


(last_key, last_val) = (None, 0)
(count, mean, M2) = (0, 0, 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, str(round(sqrt(M2/count), 2))))
        (last_key, last_val) = (key, val)
        (count, mean, M2) = update((0, 0, 0), float(val))
    else:
        (count, mean, M2) = update((count, mean, M2), float(val))
        last_key = key
if last_key:
    print('%s\t%s' % (last_key, str(round(sqrt(M2/count), 2))))
```

**result:**

```
eric@eric-VirtualBox:~$ hdfs dfs -cat /user/eric/outputB/part-00000
Albuquerque	145.35
Anaheim	141.51
Anchorage	137.12
Arlington	138.74
Atlanta131.66
Aurora 142.75
Austin  132.6
Bakersfield	161.24
Baltimore	156.49
Baton Rouge	119.82
Birmingham	155.63
```

| | |
|---|---|
| Boise | 133.11 |
| Boston | 149.35 |
| Buffalo | 151.36 |
| Chandler | 138.45 |
| Charlotte | 140.89 |
| Chesapeake | 126.01 |
| Chicago | 143.99 |
| Chula Vista | 154.27 |
| Cincinnati | 138.94 |
| Cleveland | 142.54 |
| Colorado Springs | 147.71 |
| Columbus | 145.86 |
| Corpus Christi | 136.15 |
| Dallas | 150.37 |
| Denver | 131.76 |
| Detroit | 161.75 |
| Durham | 142.4 |
| El Paso | 148.82 |
| Fort Wayne | 146.93 |
| Fort Worth | 136.55 |
| Fremont | 142.48 |
| Fresno | 138.77 |
| Garland | 139.71 |
| Gilbert | 155.16 |
| Glendale | 143.66 |
| Greensboro | 144.63 |
| Henderson | 151.35 |

| | |
|---|---|
| Hialeah | 141.06 |
| Honolulu | 130.38 |
| Houston | 128.65 |
| Indianapolis | 141.14 |
| Irvine | 132.53 |
| Irving | 146.97 |
| Jacksonville | 147.61 |
| Jersey City | 135.39 |
| Kansas City | 141.38 |
| Laredo | 138.55 |
| Las Vegas | 142.42 |
| Lexington | 146.06 |
| Lincoln | 135.74 |
| Long Beach | 142.41 |
| Los Angeles | 152.48 |
| Louisville | 133.3 |
| Lubbock | 133.52 |
| Madison | 156.93 |
| Memphis | 145.72 |
| Mesa | 126.01 |
| Miami | 141.81 |
| Milwaukee | 148.3 |
| Minneapolis | 124.7 |
| Nashville | 150.69 |
| New Orleans | 151.55 |
| New York | 130.33 |
| Newark | 141.97 |

| | |
|---|---|
| Norfolk | 148.19 |
| North Las Vegas | 153.36 |
| Oakland | 134.08 |
| Oklahoma City | 144.56 |
| Omaha | 144.09 |
| Orlando | 150.36 |
| Philadelphia | 129.64 |
| Phoenix | 142.5 |
| Pittsburgh | 164.64 |
| Plano | 136.16 |
| Portland | 147.56 |
| Raleigh | 141.99 |
| Reno | 139.49 |
| Richmond | 140.38 |
| Riverside | 143.97 |
| Rochester | 142.14 |
| Sacramento | 161.18 |
| Saint Paul | 147.12 |
| San Antonio | 132.92 |
| San Bernardino | 139.7 |
| San Diego | 142.81 |
| San Francisco | 137.7 |
| San Jose | 126.59 |
| Santa Ana | 151.96 |
| Scottsdale | 147.41 |
| Seattle | 154.1 |
| Spokane | 139.51 |

St. Louis 137.14

St. Petersburg 130.6

Stockton 141.37

Tampa 131.59

Toledo 143.93

Tucson 140.92

Tulsa 131.89

Virginia Beach 129.8

Washington 123.9

Wichita 136.08

Winston–Salem 126.26

## C

**mapper_Wei_C.py**

```
#!/usr/bin/python3
import sys



for line in sys.stdin:
    line = line.strip()
    words = line.split(' ')
    for word in words[0:7:6]:
        print(word, end = '\t')
    print()
```

**reducer_Wei_C.py**

```
#!/usr/bin/python3
import sys

(last_client, last_file) = (None, None)
total = 0
for line in sys.stdin:
	(client, file) = line.strip().split('\t')
	if last_client == client and last_file != file:
		total += 1
		(last_client, last_file) = (client, file)
	elif last_client == client and last_file == file:
		(last_client, last_file) = (client, file)
	elif not last_client:
		(last_client, last_file) = (client, file)
		total = 1
	else:
		print('%s\t%s' % (last_client, str(total)))
		(last_client, last_file) = (client, file)
		total = 1
if last_client:
	print('%s\t%s' % (last_client, str(total)))
```

**result:**

```
eric@eric-VirtualBox:~/Desktop/C$ cat log.txt | ./mapper_Wei_C.py | sort | ./reducer_Wei_C.py
101.226.166.213 1
101.226.166.249 1
101.226.168.200 1
101.255.17.34   3
103.27.238.252  1
103.53.16.6     1
104.129.1.156   4
104.144.132.2   1
104.154.90.76   1
104.156.228.120 1
104.167.102.244 1
106.3.37.223    1
107.168.140.56  1
107.173.176.148 1
108.174.24.107  1
109.106.142.176 2
109.106.142.25  2
109.106.142.85  2
109.106.143.101 2
109.106.143.118 2
109.106.143.165 2
109.106.143.202 2
109.106.143.217 2
109.106.143.62  2
109.106.143.9   2
109.111.24.89   2
109.127.128.33  2
109.127.157.49  2
109.127.182.143 2
109.127.182.52  2
109.161.103.104 2
109.161.14.201  2
109.161.7.234   2
109.161.90.4    2
109.165.51.20   2
109.167.205.32  2
109.167.207.16  2
```

(Professor, for the result of the part C, I screenshotted part of them since the answer is pretty massive and I think it's a little bit clunky to lay them out here, could you please run my code if you wanna make a double check?)

## D

**mapper_Wei_D.py**

```
#!/usr/bin/python3

import sys

import string



for line in sys.stdin:
```

```
    line = line.strip()
    words = line.split(',')
    words.insert(0, words.pop())
    for word in words:
        print(word, end='\t')
    print()
```

**reducer_Wei_D.py**

```
!/usr/bin/python3
import sys

(last_key, last_v1, last_v2, last_v3, last_v4) = (None, 0, 0, 0, 0)
count = 0
for line in sys.stdin:
    (key, v1, v2, v3, v4) = line.strip().split('\t')
    if last_key and last_key != key:
        last_v1 = str(round(float(last_v1/count), 2))
        last_v2 = str(round(float(last_v2/count), 2))
        last_v3 = str(round(float(last_v3/count), 2))
        last_v4 = str(round(float(last_v4/count), 2))
        print('%s\t%s\t%s\t%s\t%s' % (last_key, last_v1, last_v2, last_v3, last_v4))
        count = 1
        (last_key, last_v1, last_v2, last_v3, last_v4) = (key, v1, v2, v3, v4)
    else:
        count += 1
```

```
    (last_key, last_v1, last_v2, last_v3, last_v4) = (key, float(last_v1) + float(v1), float(last_v2) + float(v2), float(last_v3) + float(v3), float(last_v4) + float(v4))
if last_key:
  last_v1 = str(round(float(last_v1/count), 2))
  last_v2 = str(round(float(last_v2/count), 2))
  last_v3 = str(round(float(last_v3/count), 2))
  last_v4 = str(round(float(last_v4/count), 2))
  print('%s\t%s\t%s\t%s\t%s' % (last_key, last_v1, last_v2, last_v3, last_v4))
```

**result:**

```
eric@eric-VirtualBox:~/Desktop/D$ hdfs dfs -cat /user/eric/outputD/part-00000
Iris-setosa      5.01    3.42    1.46    0.24
Iris-versicolor  5.94    2.77    4.26    1.33
Iris-virginica   6.59    2.97    5.55    2.03
```

## Task 3

***Hadoop*** *was originated from 1997,* ***Doug Cutting*** *spent three months writing the first version of* ***Lucene****, a full text library for analyzing ordinary text with the purpose of index. In three years, he open sourced Lucene and surprisingly, many people found it useful and he received many great feedback. Then in 2001, Lucene moved to* ***Apache Software Foundation****. Later he was joined by* ***Mike Cafarella*** *for putting an effort into indexing the entire web and which derived a new web crawler called* ***Apache Nutch****. They deployed Nutch on a single machine which unexpectedly limited the total number of pages to 100 million, so that they needed a distributed storage layer for improving the scalability and handling component, till 2003, Google published the Google File System paper which perfectly could solve the very same problems, so they started to implement it in java and finished in 2004, naming it Nutch Distributed File System(****NDFS****) and which successfully helped them out by splitting file into chunks and chunks into nodes. After that, they embarked on the exploration between various data processing models, trying to figure out a way to achieve the parallelism, luckily they were enlightened again and worked out Parallelization, Distribution and Fault-tolerance. Then in 2006, Cutting created a new incubating*

*project called **Hadoop**, consisting of **Hadoop Common**, **HDFS** and **MapReduce.** Meanwhile, Yahoo! also had the same problem as theirs, and after negotiation, they agreed to replace the original system by Hadoop and worked great. In the following years, Hadoop graduate to the top level and revolutionized data storage to possibly keep all the data, getting blooming and some services were also provided by Amazon. The emergence of relational databases effectively decrease the cost of memory and although these limitations are long gone, we need to remember the enormous benefit of information about history. In 2012, MapReduce v2 named **YARN** was pulled out from MapReduce codebase and marked a turn point.*