Quiz 4: Hadoop for Fun and Profit (125 pts)

1. Functional Programming [25 points]

Write functional functions as described below:

1. Create a function `add` that takes an arbitrary number of arguments, and adds them all. Also create a function `sub` that subtracts all the arguments but the first from the first[1]. Also create a function `ra_sub` that performs right-associative subtraction

```
add(1, 2, 3) => 6
sub(5, 1, 2) => 2
ra_sub(5, 1, 2) => (5 - (1 - 2)) => 6
```

2. Create a function `zip` that takes an arbitrary number of sequences, and zips them, i.e. creates a list of lists, where the inner lists consist of the first elements from the given sequences, then the second elements from the given sequences, and so on.

```
zip([1, 2, 3], [4, 5, 6]) => [[1, 4], [2, 5], [3, 6]]
```

[1] Hint: The python operators are implemented in the Python operator library.

```
zip([1, 2, 3], [4, 5, 6], [7, 8, 9]) => [[1, 4, 7], [2, 5, 8], [3, 6, 9]]
```

3. Create a function `zipwith` that takes a function f and an arbitrary number of sequences, and returns a list of f applied to the first elements of the given sequences, followed by f applied to the second elements of the sequences, and so on.

```
zipwith(add, [1, 2, 3], [4, 5, 6]) => [5, 7, 9]
zipwith(add, [1, 2, 3], [4, 5, 6], [1, 1, 1]) => [6, 8, 10]
```

4. Create a function `flatten` that can flatten a tree.

```
flatten([1, [2, [3, 4], [5, 6], 7], 8, [9, 10]])
    => [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

5. Create a function `group_by` that takes a function and a sequence and groups the elements of the sequence based on the result of the given function. In the example below, `len` returns the length of a sequence.

```
group_by(len, ["hi", "dog", "me", "bad", "good"])
     => {2: ["hi", "me"], 3: ["dog", "bad"], 4: ["good"]}
```

2. Confirming Hadoop Installation [15 points]

1. [3 points] Acquire the cluster.[2]
   i. Process: after clicking on the create cluster button, I made sure to deselect "Configure all instances to have only internal IP addresses." under Customize Cluster. For the rest of the steps I followed the professor's instructions. After setting this up, I went under the VM tab to launch the console, which succeeded. I needed to authorize launching the console though before I could use it.
2. [3 points] Load the data into the master, move the data into HDFS.

```
nanxi_liu@cluster-44d8-m:~$ hadoop fs -mkdir /user/singhj
nanxi_liu@cluster-44d8-m:~$ ls
big-data-repo
nanxi_liu@cluster-44d8-m:~$ hadoop fs -mkdir /user/singhj/five-books
nanxi_liu@cluster-44d8-m:~$ hadoop fs -put ~/big-data-repo/five-books/* /user/singhj/five-books
nanxi_liu@cluster-44d8-m:~$ hadoop fs -ls /user/singhj/five-books
Found 5 items
-rw-r--r--   1 nanxi_liu hadoop     179903 2024-10-12 16:20 /user/singhj/five-books/a_tangled_tale.txt
-rw-r--r--   1 nanxi_liu hadoop     173379 2024-10-12 16:20 /user/singhj/five-books/alice_in_wonderland.txt
-rw-r--r--   1 nanxi_liu hadoop     394246 2024-10-12 16:20 /user/singhj/five-books/sylvie_and_bruno.txt
-rw-r--r--   1 nanxi_liu hadoop     458755 2024-10-12 16:20 /user/singhj/five-books/symbolic_logic.txt
-rw-r--r--   1 nanxi_liu hadoop     135443 2024-10-12 16:20 /user/singhj/five-books/the_game_of_logic.txt
nanxi_liu@cluster-44d8-m:~$
```

3. [3 points] *Without writing any code of your own*, verify that you have a good installation of hadoop by running wordcount on five-books. The command is similar to

hadoop jar /usr/lib/hadoop-mapreduce/hadoop-mapreduce-examples.jar wordcount /user/singhj/five-books /books-count

output:

nanxi_liu@cluster-44d8-m:~$ hadoop jar /usr/lib/hadoop-mapreduce/hadoop-mapreduce-examples.jar wordcount /user/singhj/five-books /books-count

[2] Be sure to enable the cloud-platform scope for the cluster (found under the "Manage security" menu when creating the Dataproc cluster on Compute Engine.

```
2024-10-12 16:22:46,203 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at
cluster-44d8-m.c.tufts-cs-119-438414.internal./10.128.0.3:8032
2024-10-12 16:22:46,393 INFO client.AHSProxy: Connecting to Application History server at
cluster-44d8-m.c.tufts-cs-119-438414.internal./10.128.0.3:10200
2024-10-12 16:22:46,679 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path:
/tmp/hadoop-yarn/staging/nanxi_liu/.staging/job_1728749799628_0001
2024-10-12 16:22:47,274 INFO input.FileInputFormat: Total input files to process : 5
2024-10-12 16:22:47,366 INFO mapreduce.JobSubmitter: number of splits:5
2024-10-12 16:22:48,086 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1728749799628_0001
2024-10-12 16:22:48,086 INFO mapreduce.JobSubmitter: Executing with tokens: []
2024-10-12 16:22:48,303 INFO conf.Configuration: resource-types.xml not found
2024-10-12 16:22:48,303 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2024-10-12 16:22:48,822 INFO impl.YarnClientImpl: Submitted application application_1728749799628_0001
2024-10-12 16:22:49,048 INFO mapreduce.Job: The url to track the job:
http://cluster-44d8-m.c.tufts-cs-119-438414.internal.:8088/proxy/application_1728749799628_0001/
2024-10-12 16:22:49,049 INFO mapreduce.Job: Running job: job_1728749799628_0001
2024-10-12 16:23:07,321 INFO mapreduce.Job: Job job_1728749799628_0001 running in uber mode : false
2024-10-12 16:23:07,323 INFO mapreduce.Job:  map 0% reduce 0%
2024-10-12 16:23:16,446 INFO mapreduce.Job:  map 20% reduce 0%
2024-10-12 16:23:25,529 INFO mapreduce.Job:  map 40% reduce 0%
2024-10-12 16:23:33,587 INFO mapreduce.Job:  map 60% reduce 0%
2024-10-12 16:23:40,636 INFO mapreduce.Job:  map 80% reduce 0%
2024-10-12 16:23:47,693 INFO mapreduce.Job:  map 100% reduce 0%
2024-10-12 16:23:57,768 INFO mapreduce.Job:  map 100% reduce 100%
2024-10-12 16:24:01,795 INFO mapreduce.Job: Job job_1728749799628_0001 completed successfully
2024-10-12 16:24:01,913 INFO mapreduce.Job: Counters: 55
        File System Counters
                FILE: Number of bytes read=596672
                FILE: Number of bytes written=2917863
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=1342376
                HDFS: Number of bytes written=313829
                HDFS: Number of read operations=20
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=3
                HDFS: Number of bytes read erasure-coded=0
        Job Counters
                Killed map tasks=1
                Launched map tasks=5
                Launched reduce tasks=1
                Data-local map tasks=5
                Total time spent by all maps in occupied slots (ms)=111880057
                Total time spent by all reduces in occupied slots (ms)=23122512
                Total time spent by all map tasks (ms)=34141
                Total time spent by all reduce tasks (ms)=7056
                Total vcore-milliseconds taken by all map tasks=34141
                Total vcore-milliseconds taken by all reduce tasks=7056
                Total megabyte-milliseconds taken by all map tasks=111880057
                Total megabyte-milliseconds taken by all reduce tasks=23122512
        Map-Reduce Framework
                Map input records=35119
                Map output records=219095
                Map output bytes=2091576
                Map output materialized bytes=596696
                Input split bytes=650
                Combine input records=219095
                Combine output records=42051
                Reduce input groups=29287
                Reduce shuffle bytes=596696
                Reduce input records=42051
                Reduce output records=29287
                Spilled Records=84102
                Shuffled Maps =5
                Failed Shuffles=0
                Merged Map outputs=5
                GC time elapsed (ms)=274
                CPU time spent (ms)=8900
                Physical memory (bytes) snapshot=3406417920
                Virtual memory (bytes) snapshot=28296151040
                Total committed heap usage (bytes)=3219128320
                Peak Map Physical memory (bytes)=650792960
                Peak Map Virtual memory (bytes)=4718350336
```

```
        Peak Reduce Physical memory (bytes)=387821568
        Peak Reduce Virtual memory (bytes)=4716748800
Shuffle Errors
        BAD_ID=0
        CONNECTION=0
        IO_ERROR=0
        WRONG_LENGTH=0
        WRONG_MAP=0
        WRONG_REDUCE=0
File Input Format Counters
        Bytes Read=1341726
File Output Format Counters
        Bytes Written=313829
```

4. [3 points] Run wordcount using the provided **`mapper_noll.py`** and the default reducer **`aggregate`**. The command is similar to

```
mapred streaming -file ~/big-data-repo/hadoop/mapper_noll.py -mapper mapper_noll.py \
                 -input /user/singhj/five-books -reducer aggregate \
                 -output /books-stream-count
```

```
nanxi_liu@cluster-c1e3-m:~$ hadoop fs -get /books-stream-count
nanxi_liu@cluster-c1e3-m:~$ ls -la books-stream-count/
total 112
drwxr-xr-x 2 nanxi_liu nanxi_liu   4096 Oct 13 15:20 .
drwxr-xr-x 7 nanxi_liu nanxi_liu   4096 Oct 13 15:20 ..
-rw-r--r-- 1 nanxi_liu nanxi_liu      0 Oct 13 15:20 _SUCCESS
-rw-r--r-- 1 nanxi_liu nanxi_liu 103696 Oct 13 15:20 part-00000
```

5. [3 points] Run wordcount using the provided **`mapper_noll.py`** and the provided reducer **`reducer_noll.py`**. The command is similar to

```
mapred streaming -files ~/big-data-repo/hadoop/mapper_noll.py ~/big-data-repo/hadoop/reducer_noll.py \
                         -mapper mapper_noll.py   \
                         -reducer reducer_noll.py \
                         -input /user/singhj/five-books \
                         -output /books-my-own-counts
```

```
nanxi_liu@cluster-c1e3-m:~$ hadoop fs -get /books-my-own-count
get: `/books-my-own-count': No such file or directory
nanxi_liu@cluster-c1e3-m:~$ hadoop fs -get /books-my-own-counts
nanxi_liu@cluster-c1e3-m:~$ ls -la books-my-own-counts/
total 240
drwxr-xr-x 2 nanxi_liu nanxi_liu   4096 Oct 13 15:33 .
drwxr-xr-x 8 nanxi_liu nanxi_liu   4096 Oct 13 15:33 ..
-rw-r--r-- 1 nanxi_liu nanxi_liu      0 Oct 13 15:33 _SUCCESS
-rw-r--r-- 1 nanxi_liu nanxi_liu 236309 Oct 13 15:33 part-00000
nanxi_liu@cluster-c1e3-m:~$
```

3. Analyzing Server Logs [55 points]

A dataset representing Apache web server logs is available as [access.log]. Each row has the following schema:

- **IP of client:** This refers to the IP address of the client that sent the request to the server.
- **Remote Log Name:** Remote name of the User performing the request. In the majority of the applications, this is confidential information and is hidden or not available.
- **User ID:** The ID of the user performing the request. In the majority of the applications, this is a piece of confidential information and is hidden or not available.

- **Date and Time:** The date and time of the request are represented in UTC format as follows: -Day/Month/Year:Hour:Minutes: Seconds +Time-Zone-Correction.
- **Request Type:** The type of request (GET, PUT, POST, etc.) that the server got. This depends on the operation that the request will do.
- **API:** The API of the website to which the request is related. Example: When a user accesses a cart on a shopping website, the API comes as /usr/cart.
- **Protocol and Version:** Protocol used for connecting with server and its version.
- **Status Code:** Status code that the server returned for the request. Eg: 404 is sent when a requested resource is not found. 200 is sent when the request was successfully served. *See the http status code registry listing for interpretations of status codes.*
- **Byte:** The amount of data in bytes that was sent back to the client.
- **Referrer:** The websites/source from where the user was directed to the current website. If none, it is represented by "-".
- **User Agent String:** The user agent string contains details of the browser and the host device (like the name, version, device type etc.).
- **Response Time:** The response time the server took to serve the request. This is the difference between the timestamps when the request was received and when the request was served.

Use Hadoop to perform analytics on the provided data[3].

1. Commands used
    a. git clone https://github.com/Nanxi-Flaneuse/big-data-repo
    b. hadoop fs -mkdir /user/nanxiliu
    c. hadoop fs -mkdir /user/nanxiliu/datasets
    d. hadoop fs -put ~/big-data-repo/datasets/access.log /user/nanxiliu/datasets
    e. hadoop fs -ls /user/nanxiliu/datasets
    f. Can skip
        i. hadoop jar /usr/lib/hadoop-mapreduce/hadoop-mapreduce-examples.jar wordcount /user/singhj/datasets /books-count
        ii. hadoop fs -get /books-count
        iii. ls -la books-count/
    g. mapred streaming -file ~/big-data-repo/hadoop/mapper.py -mapper mapper.py -file ~/big-data-repo/hadoop/reducer_noll.py -reducer reducer_noll.py -input /user/nanxiliu/datasets -output /prez-sent
        i. * replace the mapper.py here with different mapper for each subquestion

[3] Hint: the above questions are all variants of wordcount, for example,
- #1 requires a wordcount of the various request types, divided by the total number of requests,
- #2 requires a transformation of the response code to response type before counting, and
- #3 requires the mapper to emit only those IP addresses that have a response code of 400 or greater.

h. hadoop fs -get /request_type
i. hadoop fs -cat /response_type/part-00000

2. [6+9=15 points[4]] What is the percentage of each request type (GET, PUT, POST, etc.)?
   a.
```
nanxi_liu@cluster-7edf-m:~$ hadoop fs -cat /request_type/part-00000
LongValueSum:get        33414
LongValueSum:head       253
LongValueSum:post       44584
```
   b. Percentages
      i. GET: 33414/78251 = 0.427
      ii. HEAD: 253/78251 = 0.003
      iii. POST: 44584/78251 = 0.57
3. [6+9=15 points] What percent of the responses fall into each of the following five types?
   - Informational responses (100–199): 0
   - Successful responses (200–299): 70684/78251 = 0.903
   - Redirection messages (300–399): 2929/78251 = 0.037
   - Client error responses (400–499): 4638/78251 = 0.059
   - Server error responses (500–599): 0
   -
```
nanxi_liu@cluster-926a-m:~$ hadoop fs -cat /response_type/part-00000
LongValueSum:Client error       4638
LongValueSum:Redirection        2929
LongValueSum:Successful 70684
```
4. [9+16=25 points] What 5 IP addresses generate the most client errors?

```
nanxi_liu@cluster-926a-m:~$ hadoop fs -cat /ip_address/part-00000 |sort -k2,2nr | head -n 5
LongValueSum:173.255.176.5      2059
LongValueSum:212.9.160.24       126
LongValueSum:13.77.204.88       78
LongValueSum:51.210.243.185     58
LongValueSum:193.106.30.100     53
nanxi_liu@cluster-926a-m:~$
```

4. Presidential Speeches [15 points]

All US presidential speeches are available as a single zip in J's Github repo.

Each speech may be cleaned with this filter:

```
import requests
import re
import string
stopwords_list =
requests.get("https://gist.githubusercontent.com/rg089/35e00abf8941d72d419224cfd5b5925d
/raw/12d899b70156fd0041fa9778d657330b024b959c/stopwords.txt").content
stopwords = list(set(stopwords_list.decode().splitlines()))
```

[4] The construct a+b=c is taken to mean a points for the mapper, b points for the reducer and c points for the total.

```
def remove_stopwords(words):
    list_ = re.sub(r"[^a-zA-Z0-9]", " ", words.lower()).split())
    return [itm for itm in list_ if itm not in stopwords]

def clean_text(text):
    text = text.lower()
    text = re.sub('\[.*?\]', '', text)
    text = re.sub('[%s]' % re.escape(string.punctuation), ' ', text)
    text = re.sub('[\d\n]', ' ', text)
    return ' '.join(remove_stopwords(text))
```

The goal of this analysis is to calculate the sentiment of each president's speeches. One way to compute the sentiment of a collection of words is to take the average of their valences.[5] The AFINN-165 collection contains the valences of 3,382 English words.

Write a function **`valence (text)`** such that it takes a line of any presidential speech and returns its valence after cleaning it. It should be a functional program, conforming to the pattern:

```
def valence(text):
    return calc_valence(clean_text(text))
```

*where **`calc_valence(text)`** is a function that you write*. Be sure to test this function under any imaginable conditions, for example:

- When text is empty,
- When text is a string of non-printable characters,
- When text is a bytecode string,
- The valence code is stored in part_4/valence.py in the zip folder

The function must be in a form that we can use for testing in our environment against data that you don't have access to. The presidential speeches should be considered a representative sample.

Compute the average valence of each president's speeches according to this outline:

1. [7 points] In the mapper (which is given a sequence of lines of speeches as input):
    a. Clean each line as suggested above,
    b. Calculate the valence of each word in the line,
    c. Emit a (tab-separated) key-value pair (president, word valence) for each word in the line.[6]
2. [6 points] In the reducer (which is given all (president, word valence) key-value pairs with the same key, i.e.president):
    a. Compute the average valence of all words spoken by the president,

[5] Source.
[6] The filename that contains the lines being processed by the mapper can be obtained as `os.environ['mapreduce_map_input_file']`.

b. Emit a (tab-separated) key-value pair (president, sentiment of president's speeches).

3. Commands used:
    a. (UNZIP THE ZIP FILE AND PUT THE UNZIPPED FILES IN A DIRECTORY) unzip prez_speeches.zip -d pres_speeches
    b. hadoop fs -mkdir /user/nanxiliu/datasets/pres_speeches_untar
    c. (move files to hadoop) hadoop fs -put ~/big-data-repo/datasets/pres_speeches_untar/* /user/nanxiliu/datasets/pres_speeches_untar
    d. mapred streaming -file ~/big-data-repo/hadoop/mapper_valence.py -mapper mapper_valence.py -file ~/big-data-repo/hadoop/reducer_valence.py -reducer mapper_valence.py -input /user/nanxiliu/datasets -output /prez_1

```
nanxi_liu@cluster-e418-m:~$ hadoop fs -cat /prez_1/part-00000
LongValueSum:adams.tar.gz       0.10317460317460317
LongValueSum:arthur.tar.gz      0.07494561993276647
LongValueSum:bharrison.tar.gz   0.0785028155018218
LongValueSum:buchanan.tar.gz    0.03171868019656647
LongValueSum:bush.tar.gz        0.0785267357031613
LongValueSum:carter.tar.gz      0.06155380054049799
LongValueSum:cleveland.tar.gz   0.06792334275840402
LongValueSum:clinton.tar.gz     0.06893872505739007
LongValueSum:coolidge.tar.gz    0.11442333602802479
LongValueSum:eisenhower.tar.gz  0.14864312014182463
LongValueSum:fdroosevelt.tar.gz 0.04397671471566609
LongValueSum:fillmore.tar.gz    0.07972304648862512
LongValueSum:ford.tar.gz        0.10470021985857746
LongValueSum:garfield.tar.gz    0.08409658617818484
LongValueSum:grant.tar.gz       0.07926829268292683
LongValueSum:gwbush.tar.gz      0.060052447552447555
LongValueSum:harding.tar.gz     0.07203743941166639
LongValueSum:harrison.tar.gz    0.09357678513205087
LongValueSum:hayes.tar.gz       0.09650220913107511
LongValueSum:hoover.tar.gz      0.05835580638883483
LongValueSum:jackson.tar.gz     0.07442066975187418
LongValueSum:jefferson.tar.gz   0.07803760361210324
LongValueSum:johnson.tar.gz     0.036992465466722475
LongValueSum:jqadams.tar.gz     0.11350153670216567
LongValueSum:kennedy.tar.gz     0.08492795483638034
LongValueSum:lbjohnson.tar.gz   0.08294334362242357
LongValueSum:lincoln.tar.gz     -0.01037556627210288
LongValueSum:madison.tar.gz     0.08685121107266436
LongValueSum:mckinley.tar.gz    0.09263775033724188
LongValueSum:monroe.tar.gz      0.13225422260718925
LongValueSum:nixon.tar.gz       0.07120036513007759
LongValueSum:obama.tar.gz       0.0979752756431674
LongValueSum:pierce.tar.gz      0.06728273692489185
LongValueSum:polk.tar.gz        0.05671766821888782
LongValueSum:reagan.tar.gz      0.07234398445379185
LongValueSum:roosevelt.tar.gz   0.04866161125153931
LongValueSum:taft.tar.gz        0.08557934508816122
LongValueSum:taylor.tar.gz      0.12014292094685128
LongValueSum:truman.tar.gz      0.10568377025916798
LongValueSum:tyler.tar.gz       0.07182009310707313
LongValueSum:vanburen.tar.gz    0.07166540166979588
LongValueSum:washington.tar.gz  0.11903986981285598
LongValueSum:wilson.tar.gz      0.08778053122055077
```

[2 points] How much data, in bytes, was emitted by the mappers?

5. Hadoop Errors [15 points]

When dealing with errors in Hadoop, where the execution is distributed to hundreds of workers, an error message could end up in a log file on any of those servers. This is a scavenger hunt question. We deliberately modify the code so it would occasionally fail and look for the error message so we can find them! The provided mapper for Hadoop Streaming, mapper_noll.py, with the changed lines shown in red[7].

Run Hadoop Streaming on the five books we have been using for practice, using this modified mapper. It will fail, of course!

Screenshots of errors below:

[7]:

```
#!/usr/bin/env python
import sys, re
import random
::
::
       x = 1 / random.randint(0,99)
::
::

if __name__ == "__main__":
 main(sys.argv)
```

```
nanxi_liu@cluster-07b3-m:~/big-data-repo/hadoop$ mapred streaming -file ~/big-data-repo/hadoop/mapper_noll_modified.py  -mapper mapper_noll_modified.py   \
                -file ~/big-data-repo/hadoop/reducer_noll.py -reducer reducer_noll.py \
                -input /user/singhj/five-books \
                -output /books-my-own-counts
WARNING: HADOOP_JOB_HISTORYSERVER_OPTS has been replaced by MAPRED_HISTORYSERVER_OPTS. Using value of HADOOP_JOB_HISTORYSERVER_OPTS.
2024-10-14 19:05:02,880 WARN streaming.StreamJob: -file option is deprecated, please use generic option -files instead.
packageJobJar: [/home/nanxi_liu/big-data-repo/hadoop/mapper_noll_modified.py, /home/nanxi_liu/big-data-repo/hadoop/reducer_noll.py] [/usr/lib/hadoop/hadoop-strea
ming-3.3.6.jar] /tmp/streamjob2653192399521179O7.jar tmpDir=null
2024-10-14 19:05:04,923 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at cluster-07b3-m.c.tufts-cs-119-438414.internal./10.128.0.
9:8032
2024-10-14 19:05:05,108 INFO client.AHSProxy: Connecting to Application History server at cluster-07b3-m.c.tufts-cs-119-438414.internal./10.128.0.9:10200
2024-10-14 19:05:05,799 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at cluster-07b3-m.c.tufts-cs-119-438414.internal./10.128.0.
9:8032
2024-10-14 19:05:05,801 INFO client.AHSProxy: Connecting to Application History server at cluster-07b3-m.c.tufts-cs-119-438414.internal./10.128.0.9:10200
2024-10-14 19:05:06,074 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/nanxi_liu/.staging/job_1728932120560_0002
2024-10-14 19:05:07,078 INFO mapred.FileInputFormat: Total input files to process : 5
2024-10-14 19:05:07,551 INFO mapreduce.JobSubmitter: number of splits:5
2024-10-14 19:05:07,924 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1728932120560_0002
2024-10-14 19:05:07,924 INFO mapreduce.JobSubmitter: Executing with tokens: []
2024-10-14 19:05:08,205 INFO conf.Configuration: resource-types.xml not found
2024-10-14 19:05:08,206 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2024-10-14 19:05:08,879 INFO impl.YarnClientImpl: Submitted application application_1728932120560_0002
2024-10-14 19:05:09,091 INFO mapreduce.Job: The url to track the job: http://cluster-07b3-m.c.tufts-cs-119-438414.internal.:8088/proxy/application_1728932120560_
0002/
2024-10-14 19:05:09,096 INFO mapreduce.Job: Running job: job_1728932120560_0002
2024-10-14 19:05:26,564 INFO mapreduce.Job: Job job_1728932120560_0002 running in uber mode : false
2024-10-14 19:05:26,566 INFO mapreduce.Job:  map 0% reduce 0%
2024-10-14 19:05:34,679 INFO mapreduce.Job: Task Id : attempt_1728932120560_0002_m_000000_0, Status : FAILED
Error: java.lang.RuntimeException: PipeMapRed.waitOutputThreads(): subprocess failed with code 1
       at org.apache.hadoop.streaming.PipeMapRed.waitOutputThreads(PipeMapRed.java:326)
       at org.apache.hadoop.streaming.PipeMapRed.mapRedFinished(PipeMapRed.java:539)
       at org.apache.hadoop.streaming.PipeMapper.close(PipeMapper.java:130)
       at org.apache.hadoop.mapred.MapRunner.run(MapRunner.java:61)
       at org.apache.hadoop.streaming.PipeMapRunner.run(PipeMapRunner.java:34)
       at org.apache.hadoop.mapred.MapTask.runOldMapper(MapTask.java:465)
       at org.apache.hadoop.mapred.MapTask.run(MapTask.java:349)
       at org.apache.hadoop.mapred.YarnChild$2.run(YarnChild.java:178)
       at java.base/java.security.AccessController.doPrivileged(Native Method)
       at java.base/javax.security.auth.Subject.doAs(Subject.java:423)
       at org.apache.hadoop.security.UserGroupInformation.doAs(UserGroupInformation.java:1899)
       at org.apache.hadoop.mapred.YarnChild.main(YarnChild.java:172)
```

```
2024-10-14 19:05:42,770 INFO mapreduce.Job: Task Id : attempt_1728932120560_0002_m_000001_0, Status : FAILED
Error: java.lang.RuntimeException: PipeMapRed.waitOutputThreads(): subprocess failed with code 1
        at org.apache.hadoop.streaming.PipeMapRed.waitOutputThreads(PipeMapRed.java:326)
        at org.apache.hadoop.streaming.PipeMapRed.mapRedFinished(PipeMapRed.java:539)
        at org.apache.hadoop.streaming.PipeMapper.close(PipeMapper.java:130)
        at org.apache.hadoop.mapred.MapRunner.run(MapRunner.java:61)
        at org.apache.hadoop.streaming.PipeMapRunner.run(PipeMapRunner.java:34)
        at org.apache.hadoop.mapred.MapTask.runOldMapper(MapTask.java:465)
        at org.apache.hadoop.mapred.MapTask.run(MapTask.java:349)
        at org.apache.hadoop.mapred.YarnChild$2.run(YarnChild.java:178)
        at java.base/java.security.AccessController.doPrivileged(Native Method)
        at java.base/javax.security.auth.Subject.doAs(Subject.java:423)
        at org.apache.hadoop.security.UserGroupInformation.doAs(UserGroupInformation.java:1899)
        at org.apache.hadoop.mapred.YarnChild.main(YarnChild.java:172)

2024-10-14 19:05:50,836 INFO mapreduce.Job: Task Id : attempt_1728932120560_0002_m_000000_1, Status : FAILED
Error: java.lang.RuntimeException: PipeMapRed.waitOutputThreads(): subprocess failed with code 1
        at org.apache.hadoop.streaming.PipeMapRed.waitOutputThreads(PipeMapRed.java:326)
        at org.apache.hadoop.streaming.PipeMapRed.mapRedFinished(PipeMapRed.java:539)
        at org.apache.hadoop.streaming.PipeMapper.close(PipeMapper.java:130)
        at org.apache.hadoop.mapred.MapRunner.run(MapRunner.java:61)
        at org.apache.hadoop.streaming.PipeMapRunner.run(PipeMapRunner.java:34)
        at org.apache.hadoop.mapred.MapTask.runOldMapper(MapTask.java:465)
        at org.apache.hadoop.mapred.MapTask.run(MapTask.java:349)
        at org.apache.hadoop.mapred.YarnChild$2.run(YarnChild.java:178)
        at java.base/java.security.AccessController.doPrivileged(Native Method)
        at java.base/javax.security.auth.Subject.doAs(Subject.java:423)
        at org.apache.hadoop.security.UserGroupInformation.doAs(UserGroupInformation.java:1899)
        at org.apache.hadoop.mapred.YarnChild.main(YarnChild.java:172)

2024-10-14 19:05:59,899 INFO mapreduce.Job: Task Id : attempt_1728932120560_0002_m_000001_1, Status : FAILED
Error: java.lang.RuntimeException: PipeMapRed.waitOutputThreads(): subprocess failed with code 1
        at org.apache.hadoop.streaming.PipeMapRed.waitOutputThreads(PipeMapRed.java:326)
        at org.apache.hadoop.streaming.PipeMapRed.mapRedFinished(PipeMapRed.java:539)
        at org.apache.hadoop.streaming.PipeMapper.close(PipeMapper.java:130)
        at org.apache.hadoop.mapred.MapRunner.run(MapRunner.java:61)
        at org.apache.hadoop.streaming.PipeMapRunner.run(PipeMapRunner.java:34)
        at org.apache.hadoop.mapred.MapTask.runOldMapper(MapTask.java:465)
        at org.apache.hadoop.mapred.MapTask.run(MapTask.java:349)
        at org.apache.hadoop.mapred.YarnChild$2.run(YarnChild.java:178)
        at java.base/java.security.AccessController.doPrivileged(Native Method)
```

```
        at java.base/javax.security.auth.Subject.doAs(Subject.java:423)
        at org.apache.hadoop.security.UserGroupInformation.doAs(UserGroupInformation.java:1899)
        at org.apache.hadoop.mapred.YarnChild.main(YarnChild.java:172)

2024-10-14 19:06:07,975 INFO mapreduce.Job: Task Id : attempt_1728932120560_0002_m_000000_2, Status : FAILED
Error: java.lang.RuntimeException: PipeMapRed.waitOutputThreads(): subprocess failed with code 1
        at org.apache.hadoop.streaming.PipeMapRed.waitOutputThreads(PipeMapRed.java:326)
        at org.apache.hadoop.streaming.PipeMapRed.mapRedFinished(PipeMapRed.java:539)
        at org.apache.hadoop.streaming.PipeMapper.close(PipeMapper.java:130)
        at org.apache.hadoop.mapred.MapRunner.run(MapRunner.java:61)
        at org.apache.hadoop.streaming.PipeMapRunner.run(PipeMapRunner.java:34)
        at org.apache.hadoop.mapred.MapTask.runOldMapper(MapTask.java:465)
        at org.apache.hadoop.mapred.MapTask.run(MapTask.java:349)
        at org.apache.hadoop.mapred.YarnChild$2.run(YarnChild.java:178)
        at java.base/java.security.AccessController.doPrivileged(Native Method)
        at java.base/javax.security.auth.Subject.doAs(Subject.java:423)
        at org.apache.hadoop.security.UserGroupInformation.doAs(UserGroupInformation.java:1899)
        at org.apache.hadoop.mapred.YarnChild.main(YarnChild.java:172)

2024-10-14 19:06:16,044 INFO mapreduce.Job: Task Id : attempt_1728932120560_0002_m_000001_2, Status : FAILED
Error: java.lang.RuntimeException: PipeMapRed.waitOutputThreads(): subprocess failed with code 1
        at org.apache.hadoop.streaming.PipeMapRed.waitOutputThreads(PipeMapRed.java:326)
        at org.apache.hadoop.streaming.PipeMapRed.mapRedFinished(PipeMapRed.java:539)
        at org.apache.hadoop.streaming.PipeMapper.close(PipeMapper.java:130)
        at org.apache.hadoop.mapred.MapRunner.run(MapRunner.java:61)
        at org.apache.hadoop.streaming.PipeMapRunner.run(PipeMapRunner.java:34)
        at org.apache.hadoop.mapred.MapTask.runOldMapper(MapTask.java:465)
        at org.apache.hadoop.mapred.MapTask.run(MapTask.java:349)
        at org.apache.hadoop.mapred.YarnChild$2.run(YarnChild.java:178)
        at java.base/java.security.AccessController.doPrivileged(Native Method)
        at java.base/javax.security.auth.Subject.doAs(Subject.java:423)
        at org.apache.hadoop.security.UserGroupInformation.doAs(UserGroupInformation.java:1899)
        at org.apache.hadoop.mapred.YarnChild.main(YarnChild.java:172)

2024-10-14 19:06:25,099 INFO mapreduce.Job:  map 100% reduce 100%
2024-10-14 19:06:27,125 INFO mapreduce.Job: Job job_1728932120560_0002 failed with state FAILED due to: Task failed task_1728932120560_0002_m_000000
Job failed as tasks failed. failedMaps:1 failedReduces:0 killedMaps:0 killedReduces: 0

2024-10-14 19:06:27,270 INFO mapreduce.Job: Counters: 11
        Job Counters
                Failed map tasks=7
```

```
2024-10-14 19:06:25,099 INFO mapreduce.Job:  map 100% reduce 100%
2024-10-14 19:06:27,125 INFO mapreduce.Job: Job job_1728932120560_0002 failed with state FAILED due to: Task failed task_1728932120560_0002_m_000000
Job failed as tasks failed. failedMaps:1 failedReduces:0 killedMaps:0 killedReduces: 0

2024-10-14 19:06:27,270 INFO mapreduce.Job: Counters: 11
        Job Counters
                Failed map tasks=7
                Killed map tasks=4
                Killed reduce tasks=1
                Launched map tasks=7
                Other local map tasks=5
                Data-local map tasks=2
                Total time spent by all maps in occupied slots (ms)=158537983
                Total time spent by all reduces in occupied slots (ms)=0
                Total time spent by all map tasks (ms)=48379
                Total vcore-milliseconds taken by all map tasks=48379
                Total megabyte-milliseconds taken by all map tasks=158537983
2024-10-14 19:06:27,272 ERROR streaming.StreamJob: Job not successful!
Streaming Command Failed!
nanxi_liu@cluster-07b3-m:~/big-data-repo/hadoop$
```

[7 points] Where (what server & location) did the divide-by-zero error messages show up and how many did you find?

Server & location:

19:05:34,679

19:05:42,770

19:05:50,836

19:05:59,899

19:06:07,975

19:06:16,044

I found 6 such errors.

[8 points] How many such messages did you find? Is the count you found consistent with what you might expect from random.randint(0,99)?

I found 6 such errors. I think they are consistent because when I compared this number with the total count counted by mapper_noll and reducer_noll, it's 6/240, or roughly 1/40. 1/40 is a bit higher than 1/100 but not abnormally high. So I think this is consistent.

References:

a. https://askubuntu.com/questions/86849/how-to-unzip-a-zip-file-from-the-terminal
b.