thread: "IPC Server handler 13 on default port 40179"

13 in total.

Yi Zhou

Quiz 4

Feb. 23rd. 2023

Question 1

Text

Description automatically generated

Here is my screenshot of my code

Here is part of the result:

A picture containing text

Description automatically generated

Each section represents different iteration from 2 to 64. And result is ranked from top, with the Traffic Rank. I normalized the Traffic Rank, so that it can shows the difference better.

According to the result, the ranking of the county has #8 #32 #28 #6 #7 #24 #76 #33 #3 #56 #22 #77 leading. Compared to the inverse hardship number from the chart, the city has the least hardship is #8, #32, #6, #7, #33, #5, #24, #72, #74, #77. The difference between ranking is getting larger and larger as ranking goes. And the traffic rank does not exactly follow the inverse of the hardship. There is difference when the hardship is close. But when the difference between hardships is larger, the difference between rankings of the traffic rank is also large. So, it does not correspond to the hardship linearly, but if we divide them by tier. The traffic rank is reliable.

Question 2

There are in total 13 error messages found cause by code 1 which means the bug of dividing zero. thread: "IPC Server handler 13 on default port 40179" . The server is at IPC server handler 13 and it is located at port 40179

Question 3

I think the first option is better. Here is the math:

In the map side evaluation, there is only (president name, valence) needs to be shuffle. There are 963 speeches which means there are 963 x 20 characters, say that the valence is a number less than 100, then there are at most 963x(20 char + 16bytes) = 19260 char + 320 bytes to shuffle.

In the reduce side, even if there are no speeches needed to calculate. But there are 3382 different words. There are 43 presidents. So, there are possibly 43 x 3382 ((president, word), 1) = 43 x 3382(20 char + 8 char + 8 bytes) = 145426 x (28 + 8) = 4071928 chars + 1163408bytes to shuffle.

So, map side evaluation is better.

As the result, the highest is Obama and lbjohnson. The lowest is Lincoln and Garfield.

Here is my code:

mapper

count = 0

for line in sys.stdin:

    line = line.strip()

    words = line.split()

    for word in words:

        if aff\_dic.get(word) == None:

            continue

        else:

            count += aff\_dic.get(word)

fname = os.environ["mapreduce\_map\_input\_file"].split("\_", 1)[0]

print ('%s\t%s' % (str(fname), str(count/1000)))

reducer

#!/usr/bin/env python

import os

from sys import stdin

president\_name = None

current\_name = None

current\_count = 0

for line in stdin:

    line = line.strip()

    president\_name, count = line.split('\t', 1)

    count = int(count)

    if current\_name == president\_name:

        current\_count += count

    else:

        if current\_name:

            print ('%s\t%s' % (current\_name, current\_count))

        current\_name = president\_name

        current\_count = count

if current\_name == president\_name:

    print ('%s\t%s' % (current\_name, current\_count))