PARALLEL DISTRIBUTING COMPUTING

Assignment Title: HADOOP

Assignment No: 03

Name: Muhammad Sharjeel Akhtar

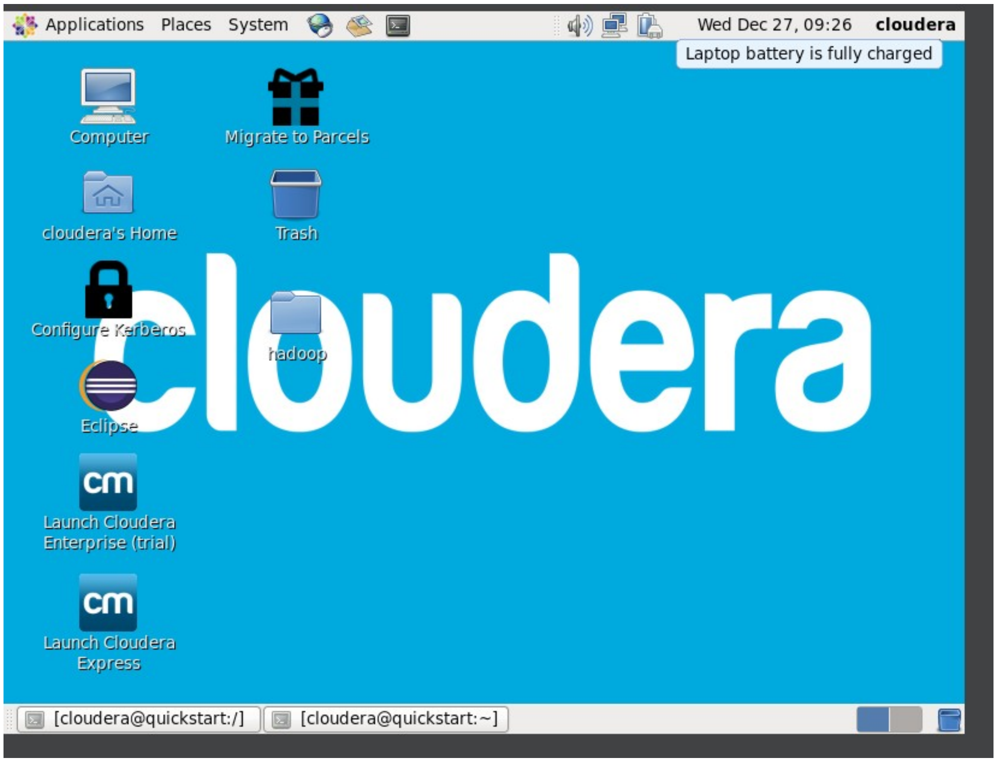
Roll No: 20P-0101

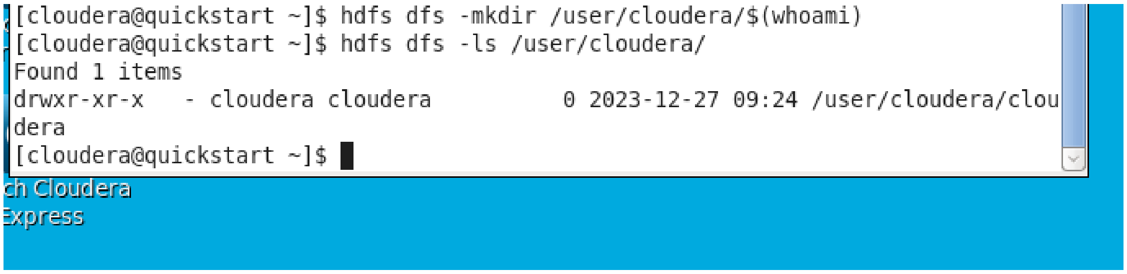
Submitted To Respected Sir: Dr OMAR USMAN Khan

Section: BCS-7F

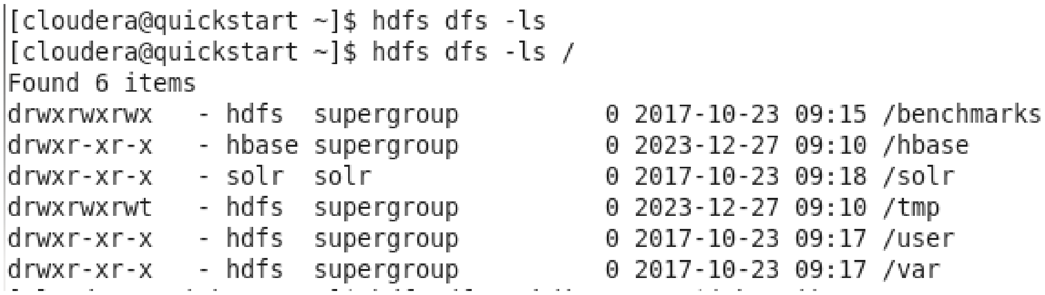
Task 1: Creating Your Directory Space

Performed this tasks using cloudera on Virtual Machine

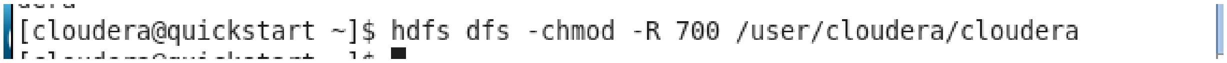




```
[cloudera@quickstart ~]$ hdfs dfs -mkdir /user/cloudera/$(whoami)
[cloudera@quickstart ~]$ hdfs dfs -ls /user/cloudera/
Found 1 items
drwxr-xr-x   - cloudera cloudera          0 2023-12-27 09:24 /user/cloudera/clou
dera
[cloudera@quickstart ~]$ 
```



```
[cloudera@quickstart ~]$ hdfs dfs -ls
[cloudera@quickstart ~]$ hdfs dfs -ls /
Found 6 items
drwxrwxrwx   - hdfs  supergroup          0 2017-10-23 09:15 /benchmarks
drwxr-xr-x   - hbase supergroup          0 2023-12-27 09:10 /hbase
drwxr-xr-x   - solr  solr                0 2017-10-23 09:18 /solr
drwxrwxrwt   - hdfs  supergroup          0 2023-12-27 09:10 /tmp
drwxr-xr-x   - hdfs  supergroup          0 2017-10-23 09:17 /user
drwxr-xr-x   - hdfs  supergroup          0 2017-10-23 09:17 /var
```



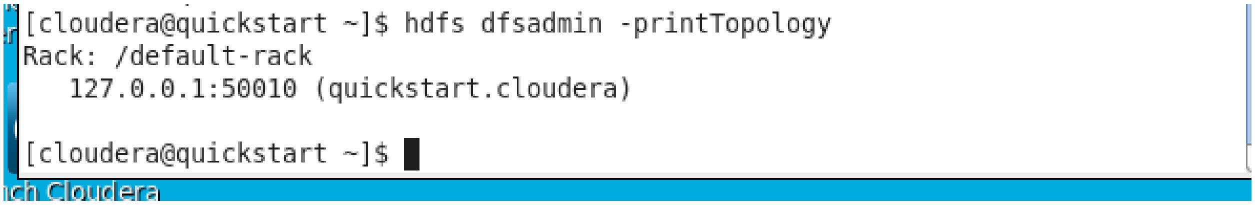
```
[cloudera@quickstart ~]$ hdfs dfs -chmod -R 700 /user/cloudera/cloudera
```

Task 2: Understanding the System

Using the following commands, address the questions:

Command : hdfs dfsadmin -printTopology

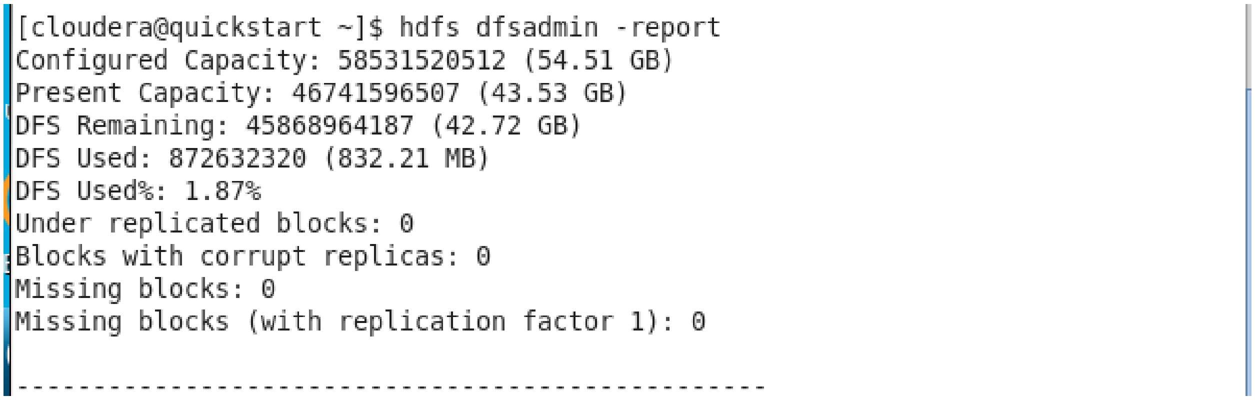
Output:



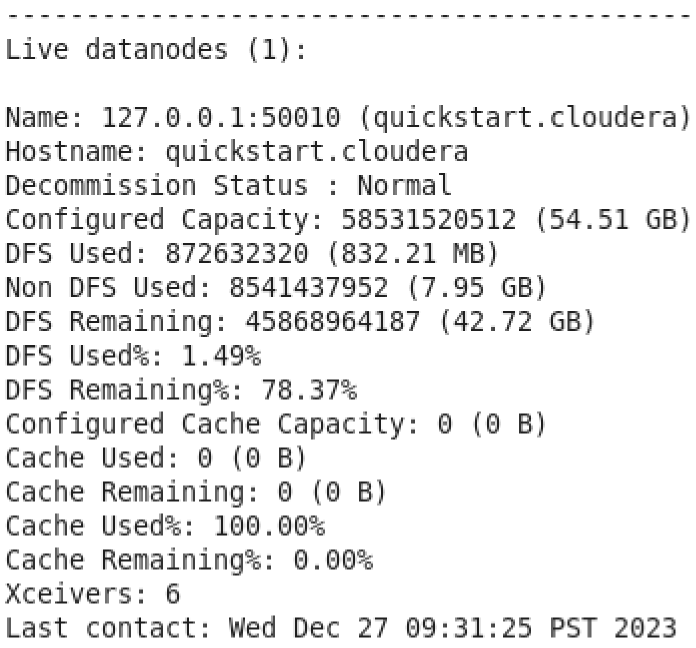
```
[cloudera@quickstart ~]$ hdfs dfsadmin -printTopology
Rack: /default-rack
   127.0.0.1:50010 (quickstart.cloudera)

[cloudera@quickstart ~]$
```

Command: hdfs dfsadmin -printTopology

Output:

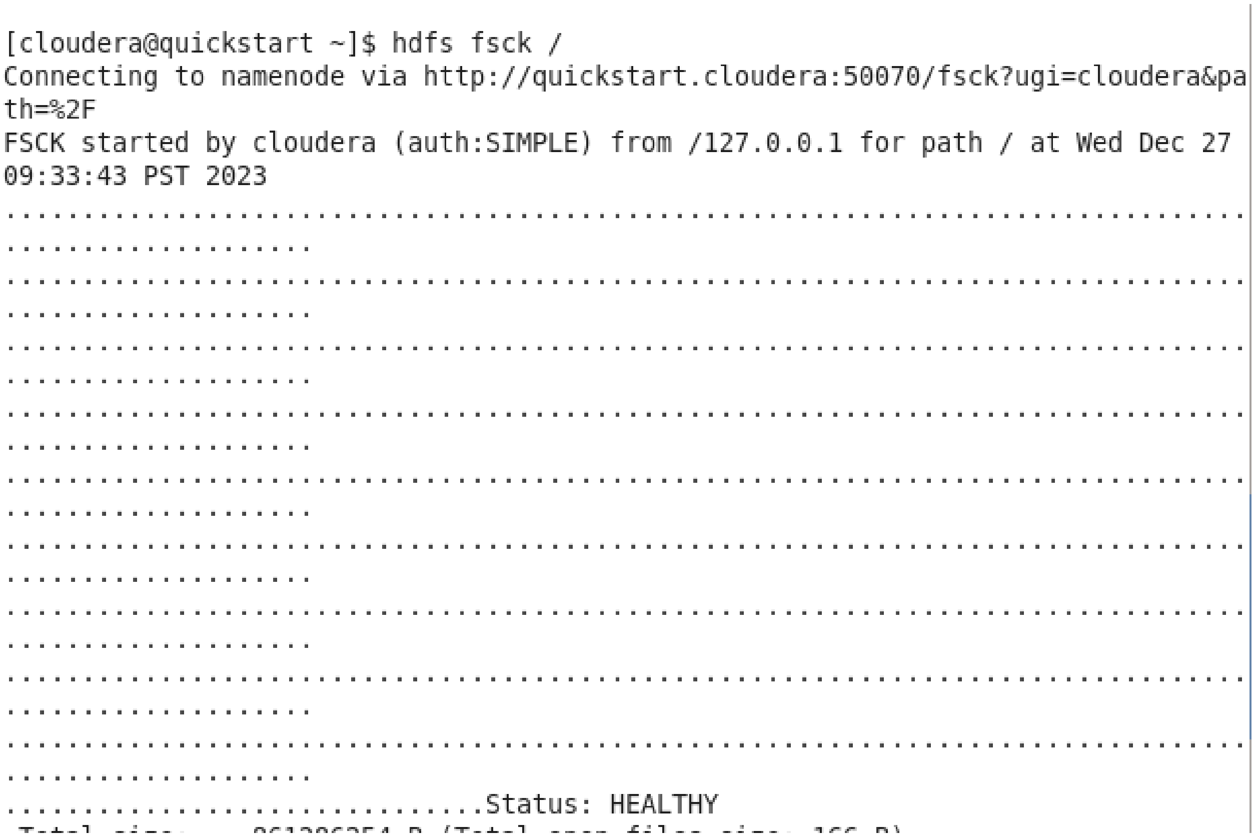


```
[cloudera@quickstart ~]$ hdfs dfsadmin -report
Configured Capacity: 58531520512 (54.51 GB)
Present Capacity: 46741596507 (43.53 GB)
DFS Remaining: 45868964187 (42.72 GB)
DFS Used: 872632320 (832.21 MB)
DFS Used%: 1.87%
Under replicated blocks: 0
Blocks with corrupt replicas: 0
Missing blocks: 0
Missing blocks (with replication factor 1): 0

-------------------------------------------------
```

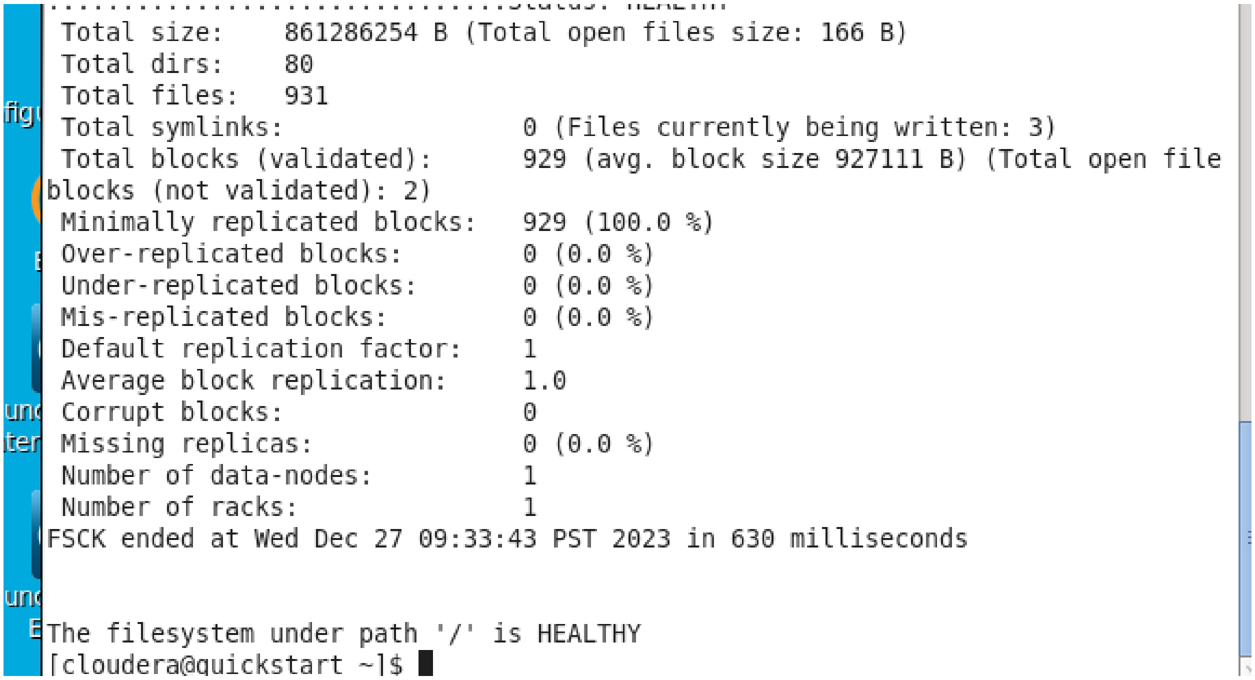


```
-------------------------------------------------
Live datanodes (1):

Name: 127.0.0.1:50010 (quickstart.cloudera)
Hostname: quickstart.cloudera
Decommission Status : Normal
Configured Capacity: 58531520512 (54.51 GB)
DFS Used: 872632320 (832.21 MB)
Non DFS Used: 8541437952 (7.95 GB)
DFS Remaining: 45868964187 (42.72 GB)
DFS Used%: 1.49%
DFS Remaining%: 78.37%
Configured Cache Capacity: 0 (0 B)
Cache Used: 0 (0 B)
Cache Remaining: 0 (0 B)
Cache Used%: 100.00%
Cache Remaining%: 0.00%
Xceivers: 6
Last contact: Wed Dec 27 09:31:25 PST 2023
```

Command: hdfs fsck /



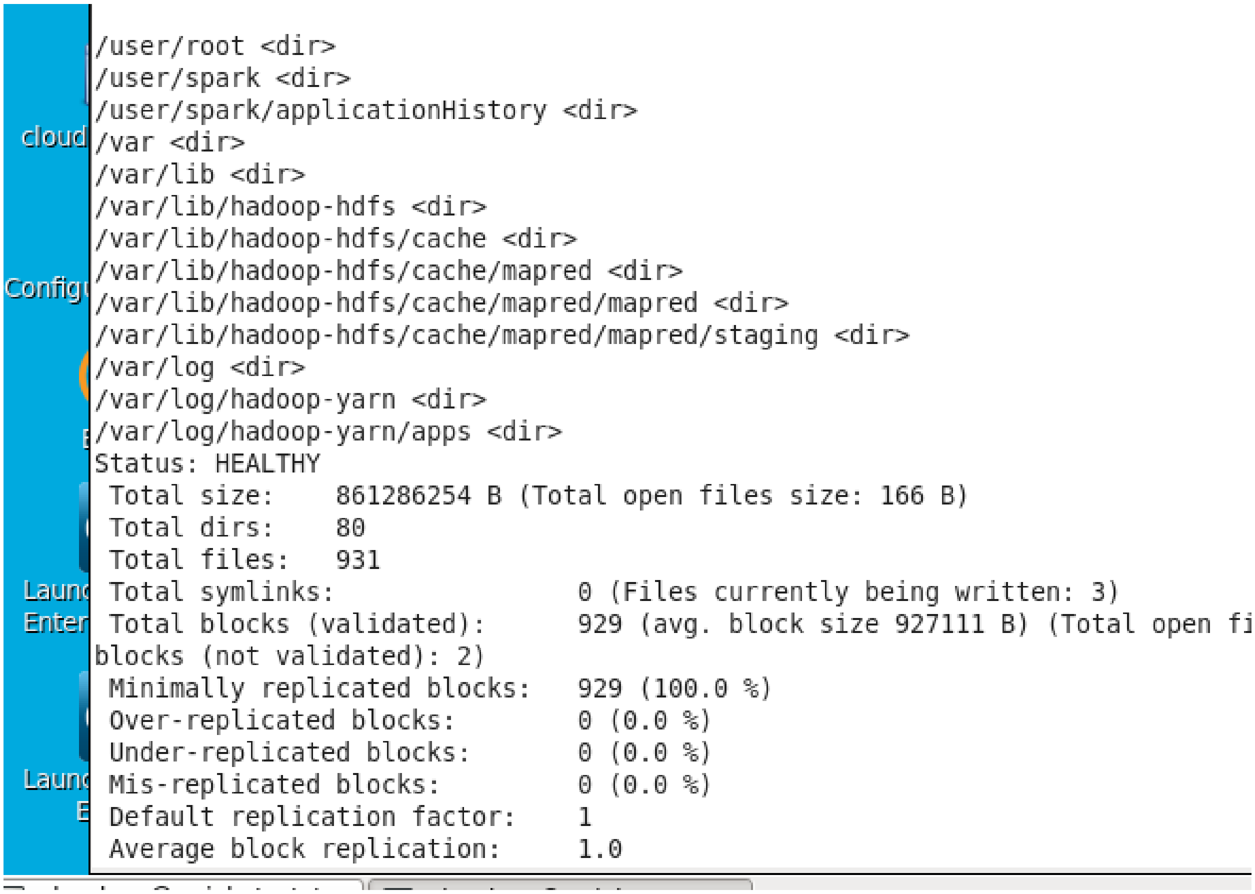
```
[cloudera@quickstart ~]$ hdfs fsck /
Connecting to namenode via http://quickstart.cloudera:50070/fsck?ugi=cloudera&pa
th=%2F
FSCK started by cloudera (auth:SIMPLE) from /127.0.0.1 for path / at Wed Dec 27
09:33:43 PST 2023
................................................................................
....................
................................................................................
....................
................................................................................
....................
................................................................................
....................
................................................................................
....................
................................................................................
....................
................................................................................
....................
................................................................................
....................
................................................................................
....................
...............................Status: HEALTHY
 Total size:    861286254 B (Total open files size: 166 B)
```



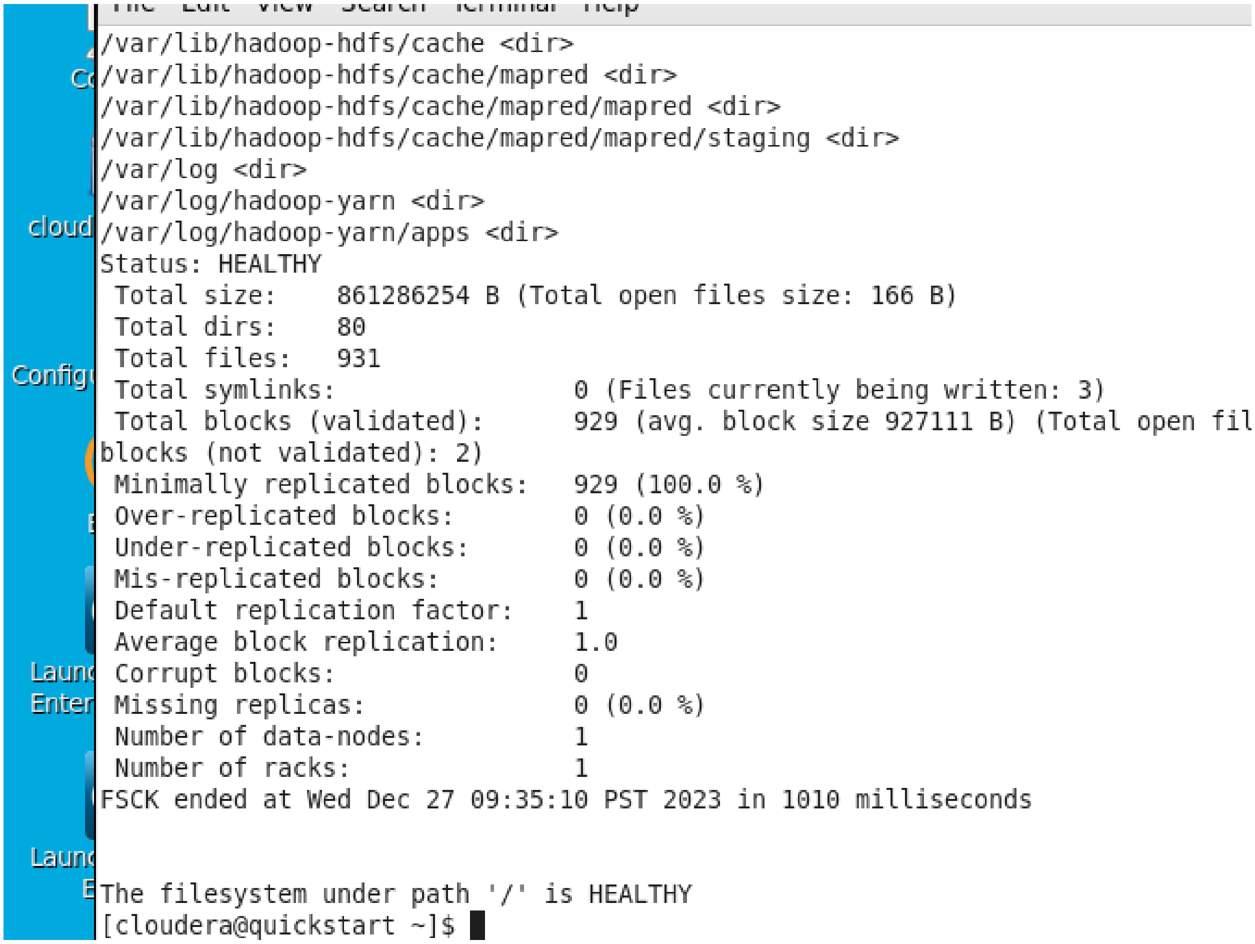
```
..................................Status: HEALTHY
 Total size:    861286254 B (Total open files size: 166 B)
 Total dirs:    80
 Total files:   931
 Total symlinks:                0 (Files currently being written: 3)
 Total blocks (validated):      929 (avg. block size 927111 B) (Total open file
blocks (not validated): 2)
 Minimally replicated blocks:   929 (100.0 %)
 Over-replicated blocks:        0 (0.0 %)
 Under-replicated blocks:       0 (0.0 %)
 Mis-replicated blocks:         0 (0.0 %)
 Default replication factor:    1
 Average block replication:     1.0
 Corrupt blocks:                0
 Missing replicas:              0 (0.0 %)
 Number of data-nodes:          1
 Number of racks:               1
FSCK ended at Wed Dec 27 09:33:43 PST 2023 in 630 milliseconds


The filesystem under path '/' is HEALTHY
[cloudera@quickstart ~]$ 
```

Command: hadoop fsck / -files -blocks -locations

Output:



```
/user/root <dir>
/user/spark <dir>
/user/spark/applicationHistory <dir>
/var <dir>
/var/lib <dir>
/var/lib/hadoop-hdfs <dir>
/var/lib/hadoop-hdfs/cache <dir>
/var/lib/hadoop-hdfs/cache/mapred <dir>
/var/lib/hadoop-hdfs/cache/mapred/mapred <dir>
/var/lib/hadoop-hdfs/cache/mapred/mapred/staging <dir>
/var/log <dir>
/var/log/hadoop-yarn <dir>
/var/log/hadoop-yarn/apps <dir>
Status: HEALTHY
 Total size:    861286254 B (Total open files size: 166 B)
 Total dirs:    80
 Total files:   931
 Total symlinks:                0 (Files currently being written: 3)
 Total blocks (validated):      929 (avg. block size 927111 B) (Total open fi
blocks (not validated): 2)
 Minimally replicated blocks:   929 (100.0 %)
 Over-replicated blocks:        0 (0.0 %)
 Under-replicated blocks:       0 (0.0 %)
 Mis-replicated blocks:         0 (0.0 %)
 Default replication factor:    1
 Average block replication:     1.0
```



```
/var/lib/hadoop-hdfs/cache <dir>
/var/lib/hadoop-hdfs/cache/mapred <dir>
/var/lib/hadoop-hdfs/cache/mapred/mapred <dir>
/var/lib/hadoop-hdfs/cache/mapred/mapred/staging <dir>
/var/log <dir>
/var/log/hadoop-yarn <dir>
/var/log/hadoop-yarn/apps <dir>
Status: HEALTHY
 Total size:    861286254 B (Total open files size: 166 B)
 Total dirs:    80
 Total files:   931
 Total symlinks:                0 (Files currently being written: 3)
 Total blocks (validated):      929 (avg. block size 927111 B) (Total open fil
blocks (not validated): 2)
 Minimally replicated blocks:   929 (100.0 %)
 Over-replicated blocks:        0 (0.0 %)
 Under-replicated blocks:       0 (0.0 %)
 Mis-replicated blocks:         0 (0.0 %)
 Default replication factor:    1
 Average block replication:     1.0
 Corrupt blocks:                0
 Missing replicas:              0 (0.0 %)
 Number of data-nodes:          1
 Number of racks:               1
FSCK ended at Wed Dec 27 09:35:10 PST 2023 in 1010 milliseconds


The filesystem under path '/' is HEALTHY
[cloudera@quickstart ~]$ 
```

Questions :

1 How many datanodes are part of the hadoop topology?

Ans :1

2 What are the IP addresses of these datanodes?

Ans: 127.0.0.1:50010

3 What is the configured and present capacity of the HDFS?

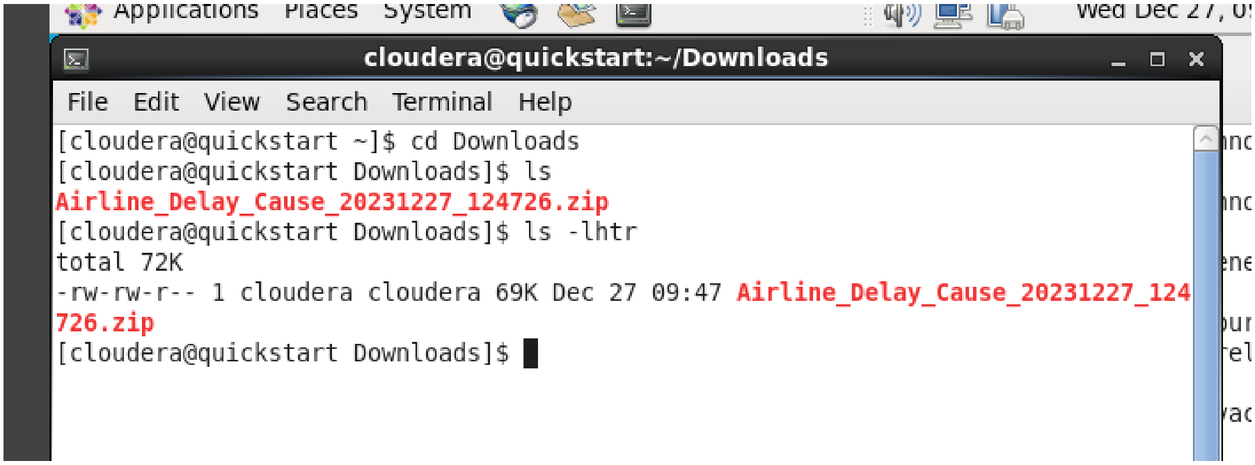
Ans: 861286254 B

4 What is the default file replication count?

Ans:1

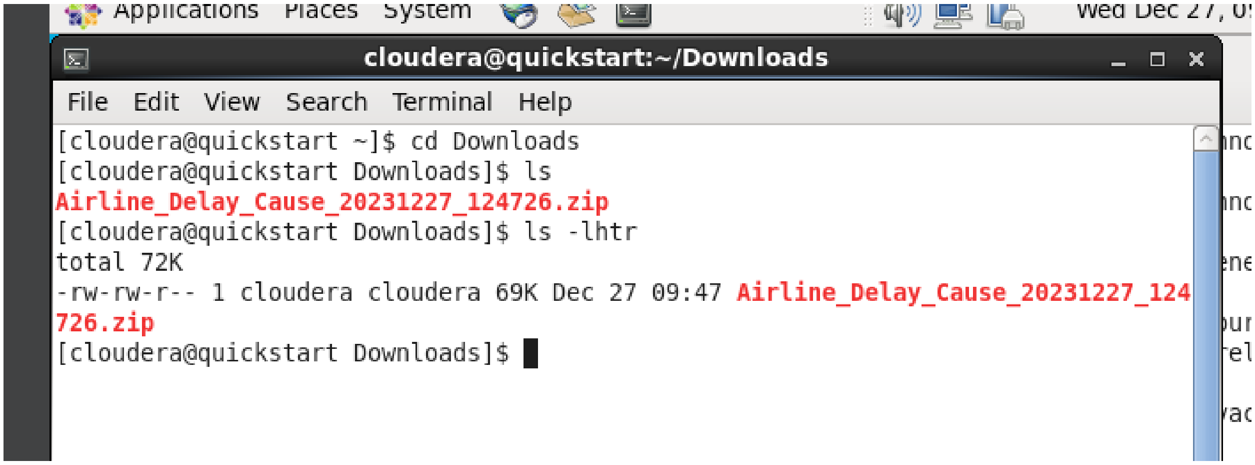
Task 3: Getting Sample Data

Data Download

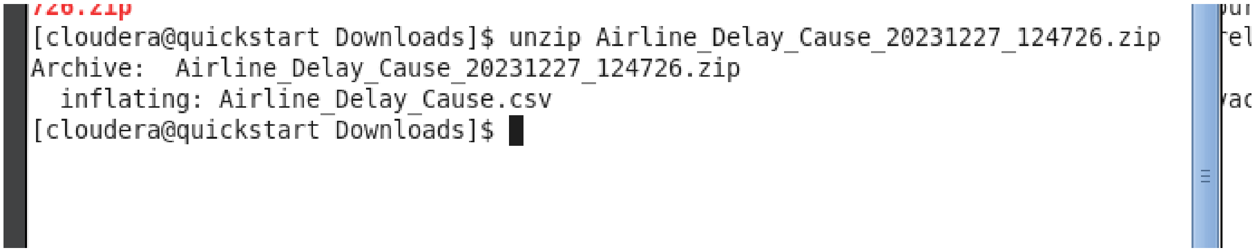

Applications Places System
Wed Dec 27, 0
cloudera@quickstart:~/Downloads
File Edit View Search Terminal Help

```
[cloudera@quickstart ~]$ cd Downloads
[cloudera@quickstart Downloads]$ ls
Airline_Delay_Cause_20231227_124726.zip
[cloudera@quickstart Downloads]$ ls -lhtr
total 72K
-rw-rw-r-- 1 cloudera cloudera 69K Dec 27 09:47 Airline_Delay_Cause_20231227_124
726.zip
[cloudera@quickstart Downloads]$ 
```

The zip file should be the last line you see. Next step, extract the zip file using the command:

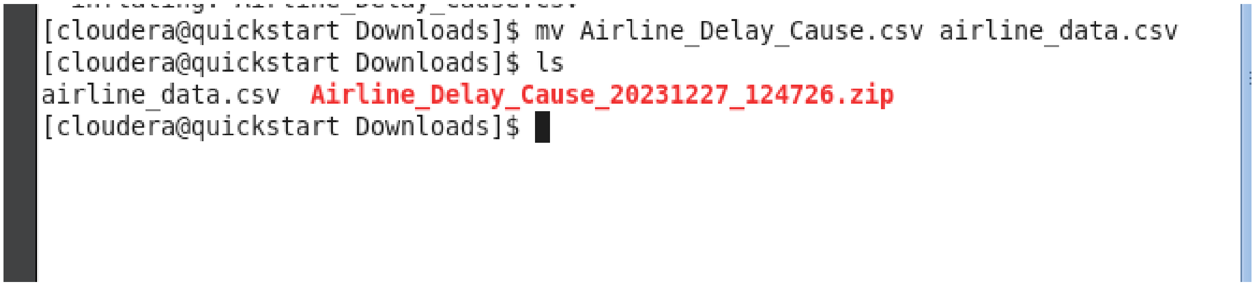

Applications Places System
Wed Dec 27, 0
cloudera@quickstart:~/Downloads
File Edit View Search Terminal Help

```
[cloudera@quickstart ~]$ cd Downloads
[cloudera@quickstart Downloads]$ ls
Airline_Delay_Cause_20231227_124726.zip
[cloudera@quickstart Downloads]$ ls -lhtr
total 72K
-rw-rw-r-- 1 cloudera cloudera 69K Dec 27 09:47 Airline_Delay_Cause_20231227_124
726.zip
[cloudera@quickstart Downloads]$ 
```



```
726.zip
[cloudera@quickstart Downloads]$ unzip Airline_Delay_Cause_20231227_124726.zip
Archive:  Airline_Delay_Cause_20231227_124726.zip
  inflating: Airline_Delay_Cause.csv
[cloudera@quickstart Downloads]$ 
```

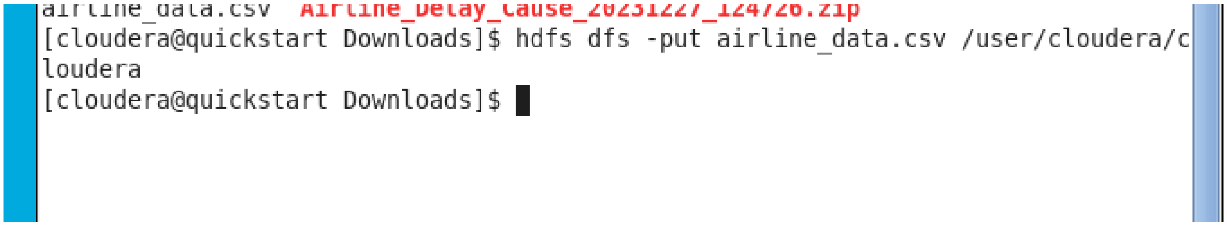
Rename the CSV file to something simpler like airline_data.csv:



```
  inflating: Airline_Delay_Cause.csv
[cloudera@quickstart Downloads]$ mv Airline_Delay_Cause.csv airline_data.csv
[cloudera@quickstart Downloads]$ ls
airline_data.csv  Airline_Delay_Cause_20231227_124726.zip
[cloudera@quickstart Downloads]$ 
```

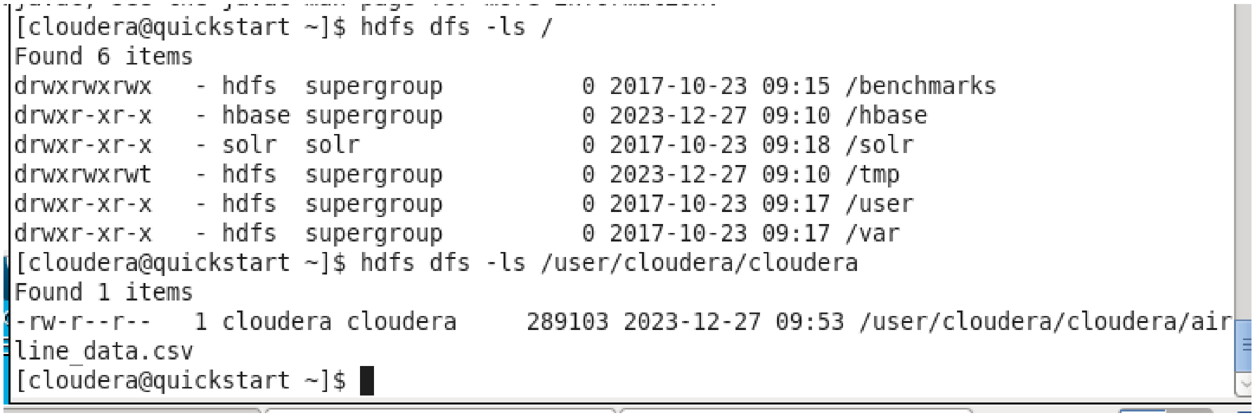
Move Data to HDFS

Copy over your data using:



```
airline_data.csv  Airline_Delay_Cause_20231227_124726.zip
[cloudera@quickstart Downloads]$ hdfs dfs -put airline_data.csv /user/cloudera/c
loudera
[cloudera@quickstart Downloads]$ 
```

Verify that it exists by:



```
[cloudera@quickstart ~]$ hdfs dfs -ls /
Found 6 items
drwxrwxrwx   - hdfs  supergroup          0 2017-10-23 09:15 /benchmarks
drwxr-xr-x   - hbase supergroup          0 2023-12-27 09:10 /hbase
drwxr-xr-x   - solr  solr                0 2017-10-23 09:18 /solr
drwxrwxrwt   - hdfs  supergroup          0 2023-12-27 09:10 /tmp
drwxr-xr-x   - hdfs  supergroup          0 2017-10-23 09:17 /user
drwxr-xr-x   - hdfs  supergroup          0 2017-10-23 09:17 /var
[cloudera@quickstart ~]$ hdfs dfs -ls /user/cloudera/cloudera
Found 1 items
-rw-r--r--   1 cloudera cloudera     289103 2023-12-27 09:53 /user/cloudera/cloudera/air
line_data.csv
[cloudera@quickstart ~]$ 
```

Question Answer

1 What is the default block size (in Mb) of the airline_data.csv file?

Ans: block size 289103 B

2 Is there any missing replicas for the file airline_data.csv?

Ans: Missing Replicas : 0

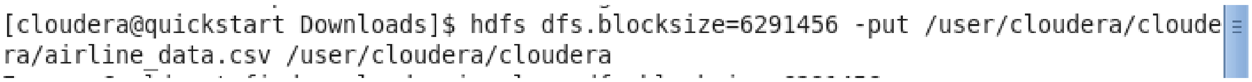
3 What command will you use to change this block size to 6 Mb (remember to convert into bytes)

Ans : bytes 6MB×1024KB/MB×1024bytes/KB=6291456bytes

So, the command to set the Hadoop block size to 6 MB in bytes would be:

bash Copy code

hadoop fs -D dfs.blocksize=6291456 -put <your-input-file> <your-output-directory>



```
[cloudera@quickstart Downloads]$ hdfs dfs.blocksize=6291456 -put /user/cloudera/cloude
ra/airline_data.csv /user/cloudera/cloudera
```

4 How many blocks are used by airline_data.csv after changing block size in Question 2?

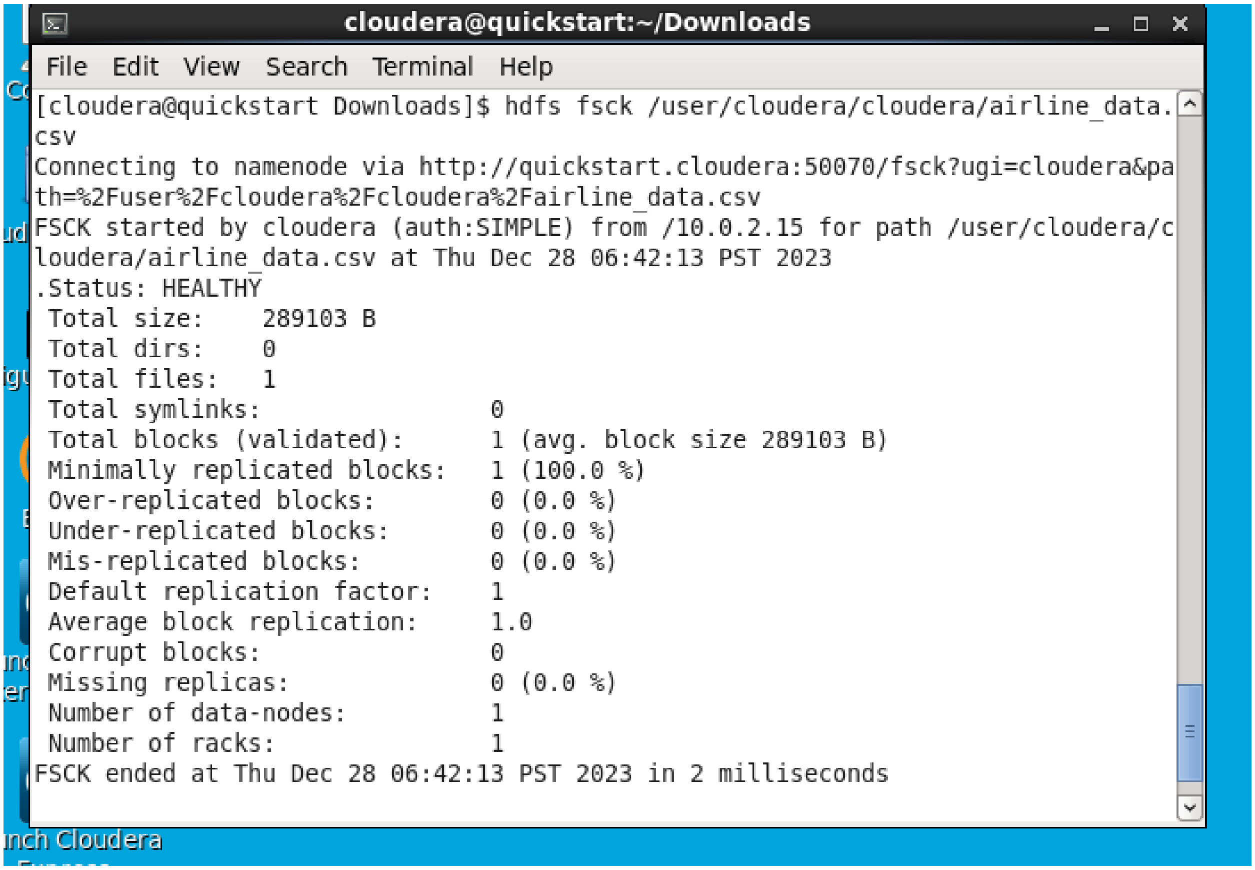
Ans : 2

5 How many missing replicas are there for file airline_data.csv after block change?

Ans : 0

6 Why are there missing replicas?

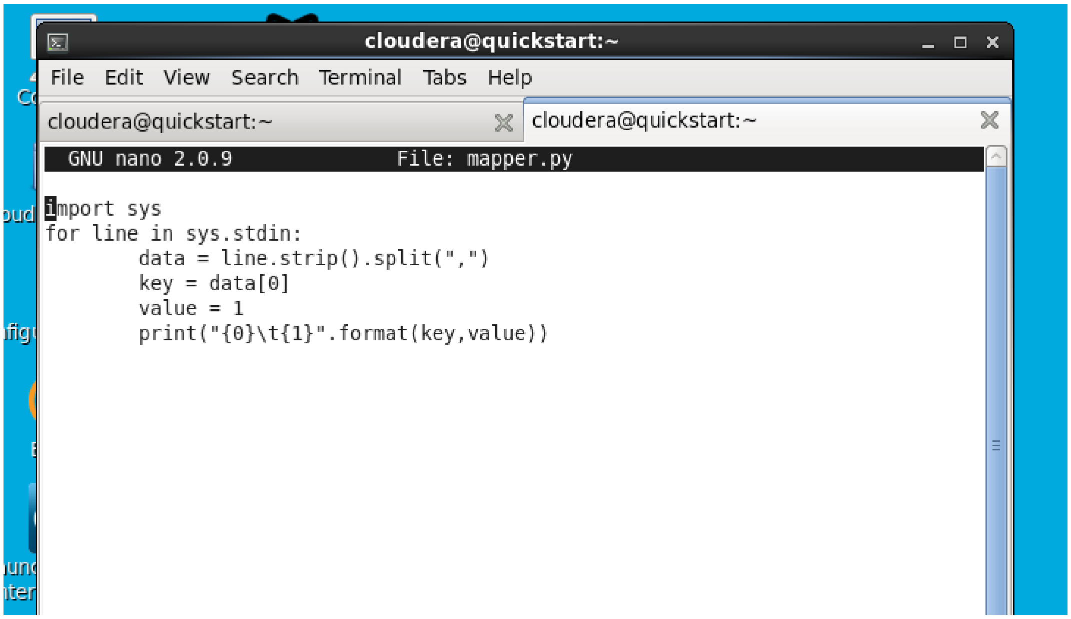
Ans: Missing replicas in Hadoop may occur due to factors such as Data Node failures, under-replicated blocks, network issues, maintenance activities, configuration errors, or manual intervention. Hadoop's automatic recovery mechanisms, like replication and balancing, are designed to address these issues, but monitoring and proper configuration are essential for a well-functioning cluster.


cloudera@quickstart:~/Downloads
File Edit View Search Terminal Help
nch Cloudera

```
[cloudera@quickstart Downloads]$ hdfs fsck /user/cloudera/cloudera/airline_data.
csv
Connecting to namenode via http://quickstart.cloudera:50070/fsck?ugi=cloudera&pa
th=%2Fuser%2Fcloudera%2Fcloudera%2Fairline_data.csv
FSCK started by cloudera (auth:SIMPLE) from /10.0.2.15 for path /user/cloudera/c
loudera/airline_data.csv at Thu Dec 28 06:42:13 PST 2023
.Status: HEALTHY
 Total size:    289103 B
 Total dirs:    0
 Total files:   1
 Total symlinks:                0
 Total blocks (validated):      1 (avg. block size 289103 B)
 Minimally replicated blocks:   1 (100.0 %)
 Over-replicated blocks:        0 (0.0 %)
 Under-replicated blocks:       0 (0.0 %)
 Mis-replicated blocks:         0 (0.0 %)
 Default replication factor:    1
 Average block replication:     1.0
 Corrupt blocks:                0
 Missing replicas:              0 (0.0 %)
 Number of data-nodes:          1
 Number of racks:               1
FSCK ended at Thu Dec 28 06:42:13 PST 2023 in 2 milliseconds
```

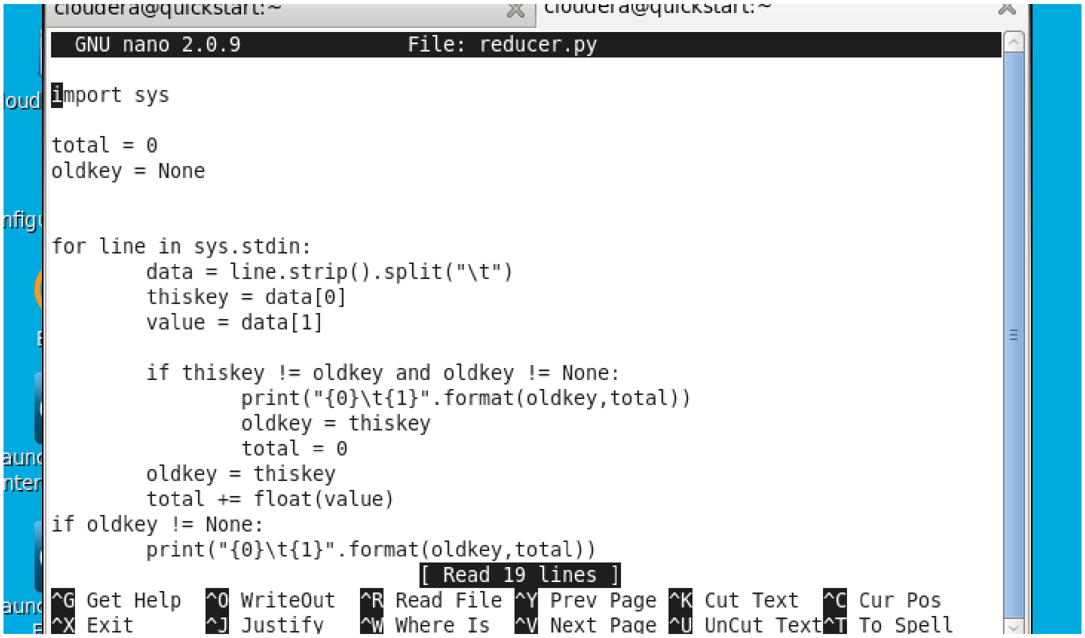
Task 4: Setting up First Map Reduce Job

Mapper.py code



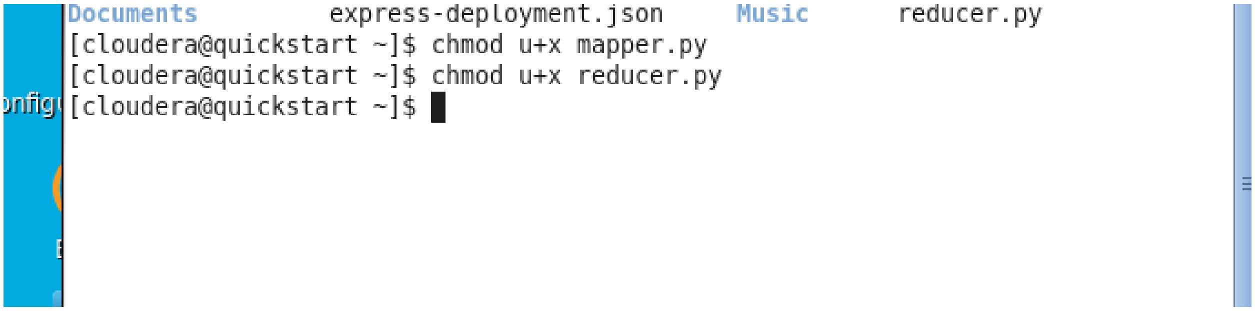
```
import sys
for line in sys.stdin:
        data = line.strip().split(",")
        key = data[0]
        value = 1
        print("{0}\t{1}".format(key,value))
```

Reducer.py code



```
import sys

total = 0
oldkey = None


for line in sys.stdin:
        data = line.strip().split("\t")
        thiskey = data[0]
        value = data[1]

        if thiskey != oldkey and oldkey != None:
                print("{0}\t{1}".format(oldkey,total))
                oldkey = thiskey
                total = 0
        oldkey = thiskey
        total += float(value)
if oldkey != None:
        print("{0}\t{1}".format(oldkey,total))
```

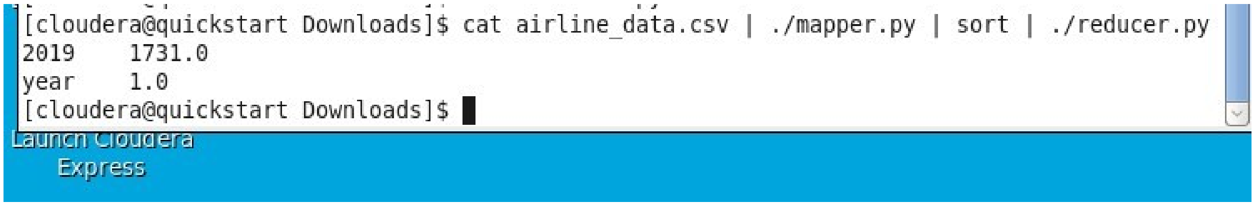
Give both the mapper.py and Reducer.py executable permissions:



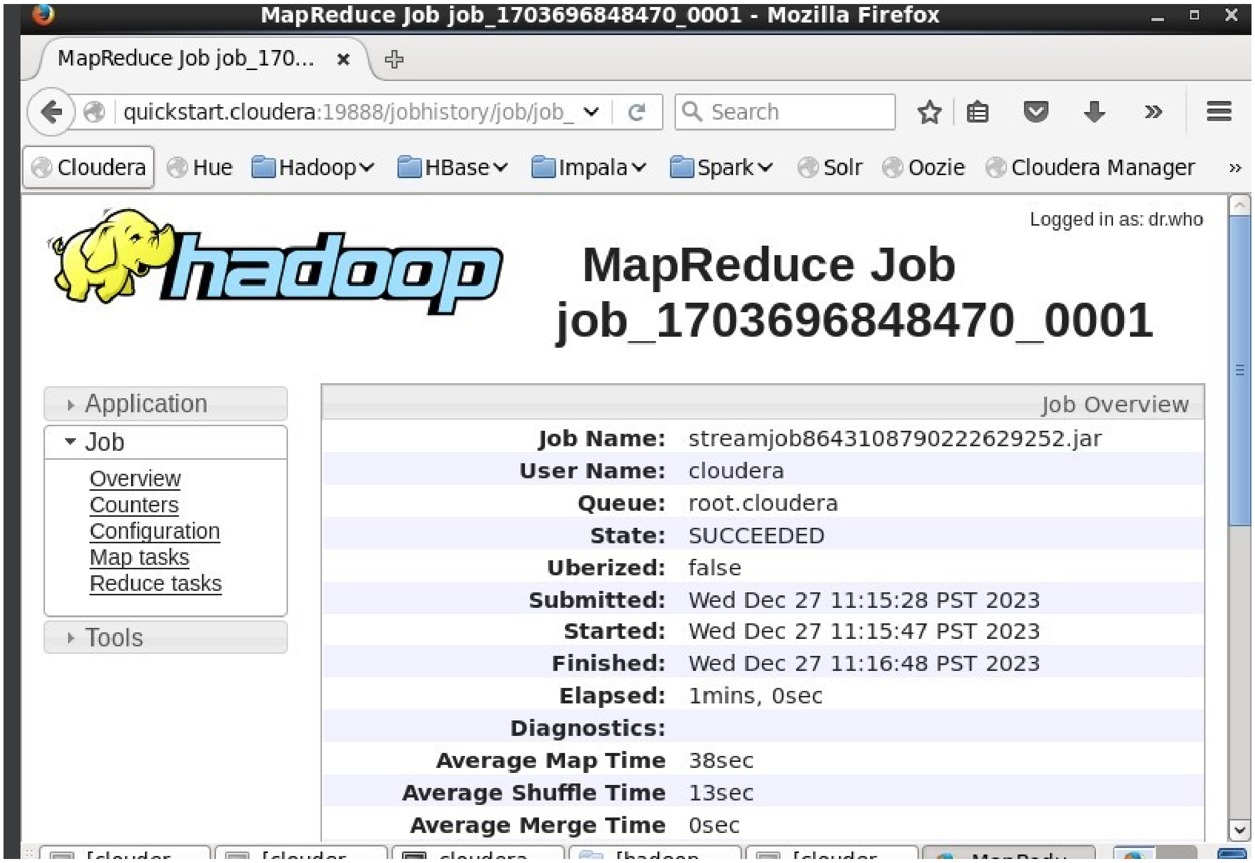
```
Documents          express-deployment.json     Music       reducer.py
[cloudera@quickstart ~]$ chmod u+x mapper.py
[cloudera@quickstart ~]$ chmod u+x reducer.py
[cloudera@quickstart ~]$ 
```

Testing Locally

Test the mapper and reducer on your local directory first, without map reduce:



```
[cloudera@quickstart Downloads]$ cat airline_data.csv | ./mapper.py | sort | ./reducer.py
2019    1731.0
year    1.0
[cloudera@quickstart Downloads]$ 
```



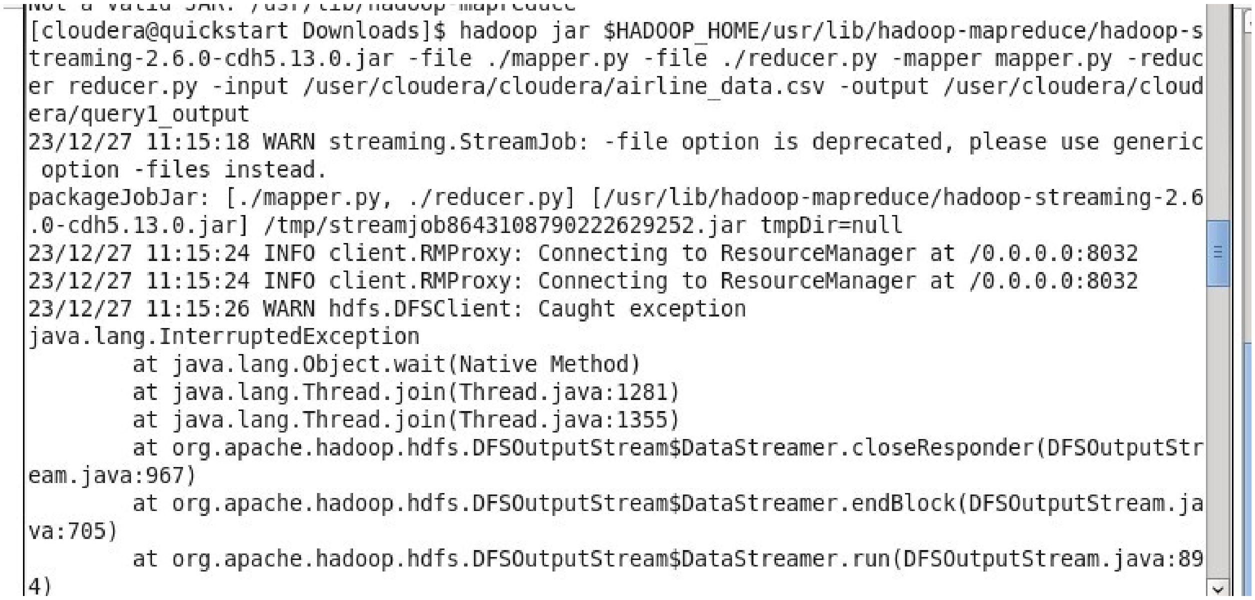
MapReduce Job job_1703696848470_0001

Job Overview

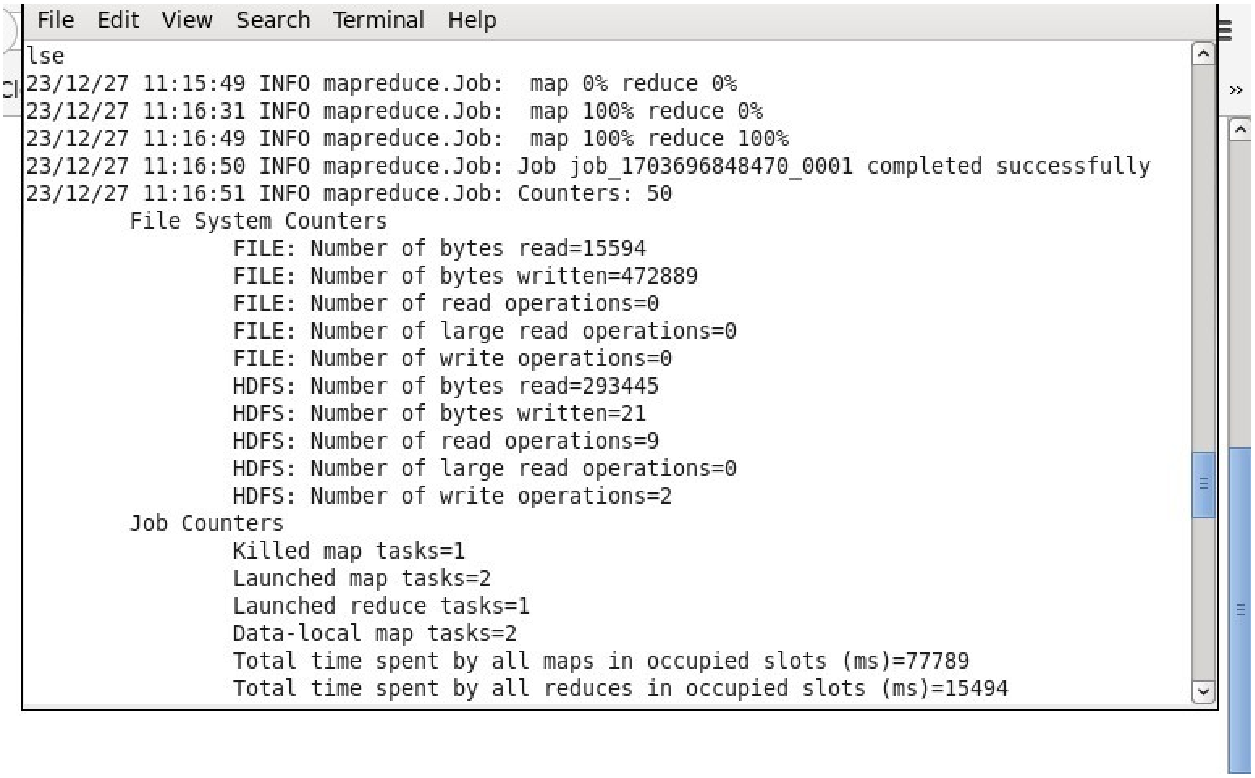
Job Name:	streamjob8643108790222629252.jar
User Name:	cloudera
Queue:	root.cloudera
State:	SUCCEEDED
Uberized:	false
Submitted:	Wed Dec 27 11:15:28 PST 2023
Started:	Wed Dec 27 11:15:47 PST 2023
Finished:	Wed Dec 27 11:16:48 PST 2023
Elapsed:	1mins, 0sec
Diagnostics:	
Average Map Time	38sec
Average Shuffle Time	13sec
Average Merge Time	0sec

Testing on Hadoop

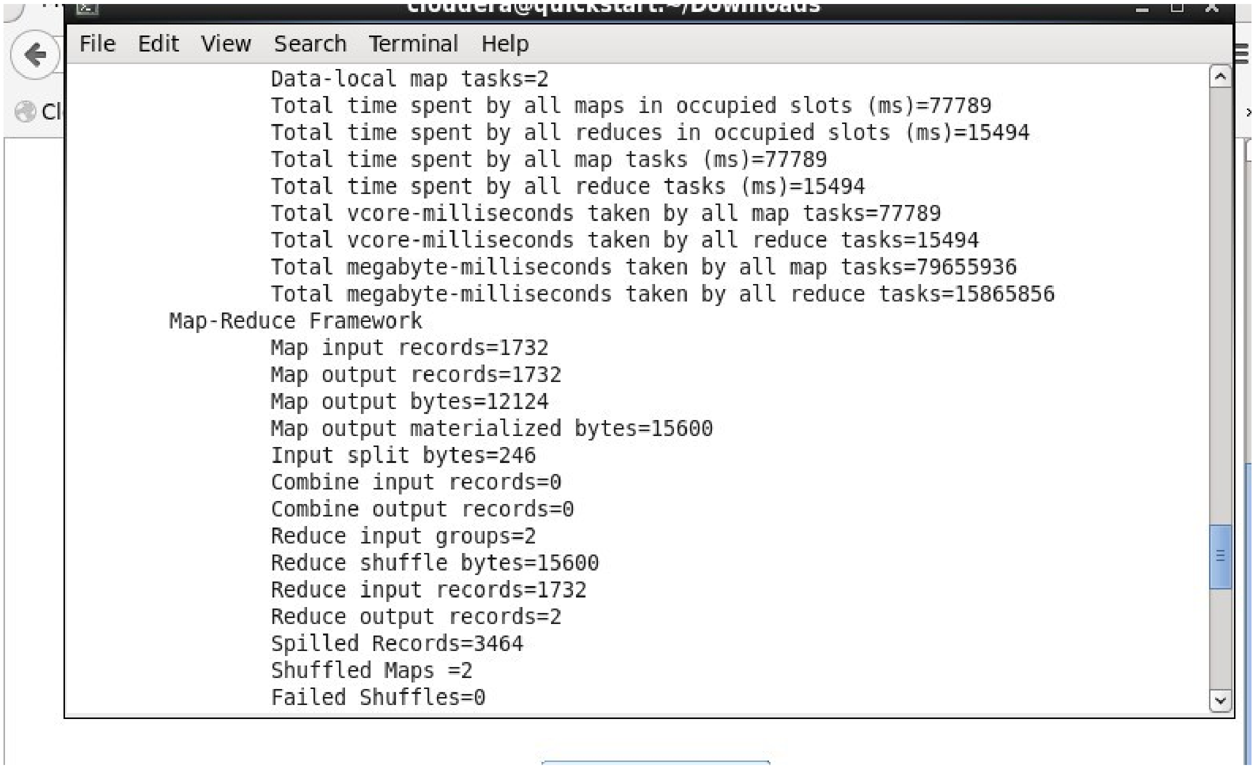
Test the mapper and reducer using hadoop:



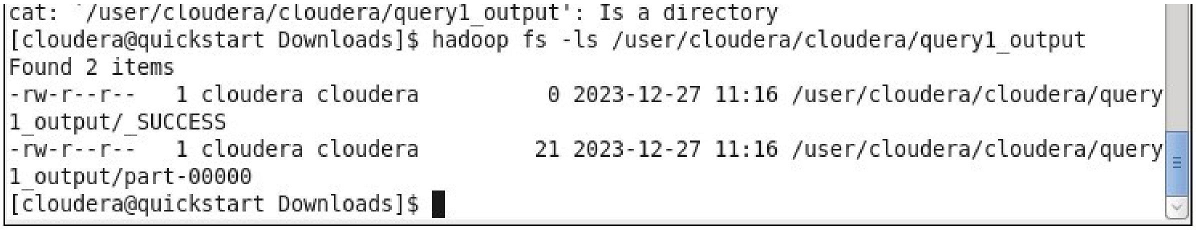
```
Not a valid JAR: /usr/lib/hadoop-mapreduce
[cloudera@quickstart Downloads]$ hadoop jar $HADOOP_HOME/usr/lib/hadoop-mapreduce/hadoop-s
treaming-2.6.0-cdh5.13.0.jar -file ./mapper.py -file ./reducer.py -mapper mapper.py -reduc
er reducer.py -input /user/cloudera/cloudera/airline_data.csv -output /user/cloudera/cloud
era/query1_output
23/12/27 11:15:18 WARN streaming.StreamJob: -file option is deprecated, please use generic
 option -files instead.
packageJobJar: [./mapper.py, ./reducer.py] [/usr/lib/hadoop-mapreduce/hadoop-streaming-2.6
.0-cdh5.13.0.jar] /tmp/streamjob8643108790222629252.jar tmpDir=null
23/12/27 11:15:24 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
23/12/27 11:15:24 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
23/12/27 11:15:26 WARN hdfs.DFSClient: Caught exception
java.lang.InterruptedException
        at java.lang.Object.wait(Native Method)
        at java.lang.Thread.join(Thread.java:1281)
        at java.lang.Thread.join(Thread.java:1355)
        at org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.closeResponder(DFSOutputStr
eam.java:967)
        at org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.endBlock(DFSOutputStream.ja
va:705)
        at org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.run(DFSOutputStream.java:89
4)
```



```
lse
23/12/27 11:15:49 INFO mapreduce.Job:  map 0% reduce 0%
23/12/27 11:16:31 INFO mapreduce.Job:  map 100% reduce 0%
23/12/27 11:16:49 INFO mapreduce.Job:  map 100% reduce 100%
23/12/27 11:16:50 INFO mapreduce.Job: Job job_1703696848470_0001 completed successfully
23/12/27 11:16:51 INFO mapreduce.Job: Counters: 50
        File System Counters
                FILE: Number of bytes read=15594
                FILE: Number of bytes written=472889
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=293445
                HDFS: Number of bytes written=21
                HDFS: Number of read operations=9
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
        Job Counters
                Killed map tasks=1
                Launched map tasks=2
                Launched reduce tasks=1
                Data-local map tasks=2
                Total time spent by all maps in occupied slots (ms)=77789
                Total time spent by all reduces in occupied slots (ms)=15494
```



```
                Data-local map tasks=2
                Total time spent by all maps in occupied slots (ms)=77789
                Total time spent by all reduces in occupied slots (ms)=15494
                Total time spent by all map tasks (ms)=77789
                Total time spent by all reduce tasks (ms)=15494
                Total vcore-milliseconds taken by all map tasks=77789
                Total vcore-milliseconds taken by all reduce tasks=15494
                Total megabyte-milliseconds taken by all map tasks=79655936
                Total megabyte-milliseconds taken by all reduce tasks=15865856
        Map-Reduce Framework
                Map input records=1732
                Map output records=1732
                Map output bytes=12124
                Map output materialized bytes=15600
                Input split bytes=246
                Combine input records=0
                Combine output records=0
                Reduce input groups=2
                Reduce shuffle bytes=15600
                Reduce input records=1732
                Reduce output records=2
                Spilled Records=3464
                Shuffled Maps =2
                Failed Shuffles=0
```



```
cat: `/user/cloudera/cloudera/query1_output': Is a directory
[cloudera@quickstart Downloads]$ hadoop fs -ls /user/cloudera/cloudera/query1_output
Found 2 items
-rw-r--r--   1 cloudera cloudera          0 2023-12-27 11:16 /user/cloudera/cloudera/query
1_output/_SUCCESS
-rw-r--r--   1 cloudera cloudera         21 2023-12-27 11:16 /user/cloudera/cloudera/query
1_output/part-00000
[cloudera@quickstart Downloads]$ 
```

Question

1 What was the <key,value> pair used in this query?

Key, Value: 2019, 1731.0 year, 1.0

How many mapper threads used?

ANS: 2 mapper threads are used

How many reducers threads used?

ANS: 1 reducer threads are used

What was the time spent by all mapper threads?

ANS: total time spent by all map tasks: 77789 ms

5 What was the time spent by all reducer threads?

ANS: total time spent by all map tasks: 15494 ms

6 What is the file name in which your output is located?

ANS: /user/cloudera/cloudera/query1_output

Variation 1

For this task, you need to calculate execution time (mapper + reducer) by two variations:

1) play with block size of airline_data.csv using the “-D dfs.blocksize=<>” argument.

2) Play with thread variation using the “-D mapred.reduce.tasks=<>”, or the “-jobconf mapred.reduce.tasks=<>” argument.

# of Reducer Tasks		# of block size		
	2	**4**	**8**	**16**
2	**0m55.481s**	**0m56.993s**	**0m56.989**	**0m57.171**
4	**1m31.392s**	**1m31.106s**	**1m15.288s**	**1m17.178s**
8	**2m20.816s**	**2m8.526s**	**2m13.253s**	**2m5.728s**
16	**3m21.319s**	**3m45.676**	**2m53.510s**	**2m55.018s**

Question Answer

1 How many output files are produced for 16 reducer

threads.

Ans : 16 output files are produced for 16 reducer threads

2 Why are some output files having 0 byte size?

Ans : Output files with 0-byte size in a Hadoop MapReduce job can be caused by: Reducer Did Not Receive Data: Uneven data distribution or partitioning issues may lead to some reducers not receiving data. Skewed Data Distribution: Skewed data, where certain keys have significantly more data, can result in some reducers having little or no data. Reducer Logic Issues: Issues in reducer logic might cause it to produce empty output, resulting in 0-byte files. Empty Output for Some Keys: If reducer logic generates empty output for certain keys, it can lead to 0-byte files.

Variation 2

For this task, you need to calculate execution time (mapper + reducer) by two variations:

1) play with block size of airline_data.csv using the “-D dfs.blocksize=<>” argument.

2) Play with thread variation using the “-D mapred.map.tasks=<>”, or the “-jobconf

mapred.map.tasks=<>” argument.

# of Map Tasks		# of block size		
	2	**4**	**8**	**16**
2	**0m48.339s**	**0m47.612s**	**0m48.121s**	**0m48.816s**
4	**1m3.852s**	**1m2.836s**	**1m1.575s**	**1m2.467s**

8	**2m7.126s**	**2m29.984s**	**1m45.561s**	**1m35.970s**
16	**2m39.526s**	**3m14.034s**	**2m40.7823s**	**2m39.739s**

Variation 3

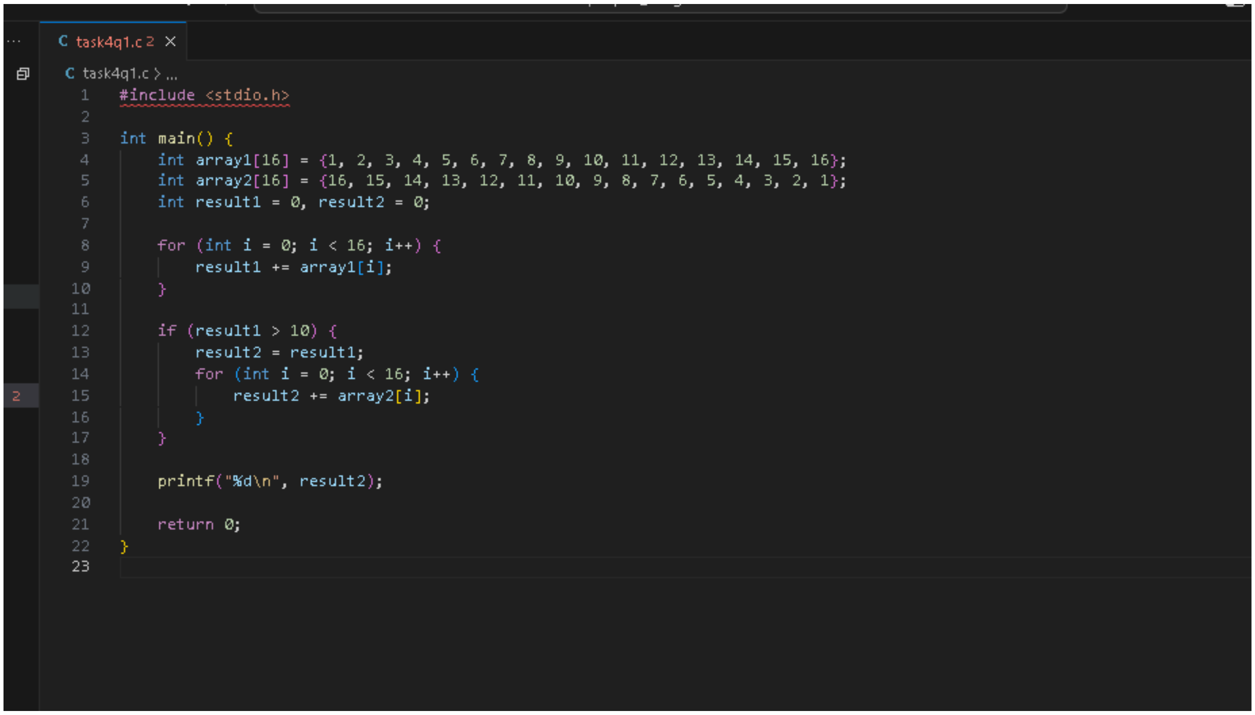
From the Variation 1 or Variation 2, choose the airline_data.csv block size which is giving best performance. and then, for this task, you need to calculate execution time (mapper + reducer) by two variations:

1) Play with thread variation using the “-D mapred.reduce.tasks=<>”, or the “-jobconf mapred.reduce.tasks=<>” argument.

2) Play with thread variation using the “-D mapred.map.tasks=<>”, or the “-jobconf mapred.map.tasks=<>” argument.

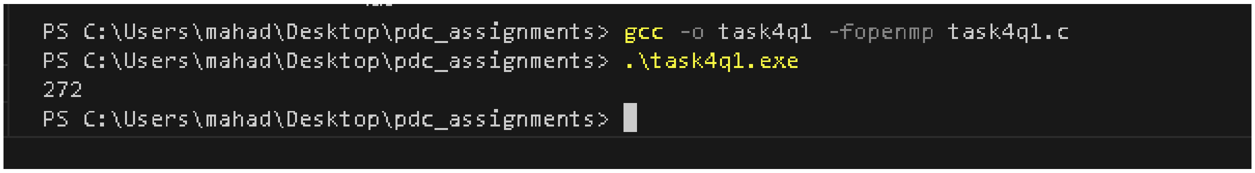
# of Map Tasks		# of Reducer Tasks		
	2	**4**	**8**	**16**
2	**1m8.676s**	**1m32.515s**	**2m6.185s**	**3m54.471s**
4	**1m23.501s**	**1m58.632s**	**2m46.928s**	**4m15.635s**
8	**2m14.683s**	**2m28.847s**	**2m54.861s**	**4m37.770s**
16	**3m10.423s**	**3m52.590s**	**4m32.098s**	**5m23.564s**

1. and array2.Code:



```c
#include <stdio.h>

int main() {
    int array1[16] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16};
    int array2[16] = {16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1};
    int result1 = 0, result2 = 0;

    for (int i = 0; i < 16; i++) {
        result1 += array1[i];
    }

    if (result1 > 10) {
        result2 = result1;
        for (int i = 0; i < 16; i++) {
            result2 += array2[i];
        }
    }

    printf("%d\n", result2);

    return 0;
}
```

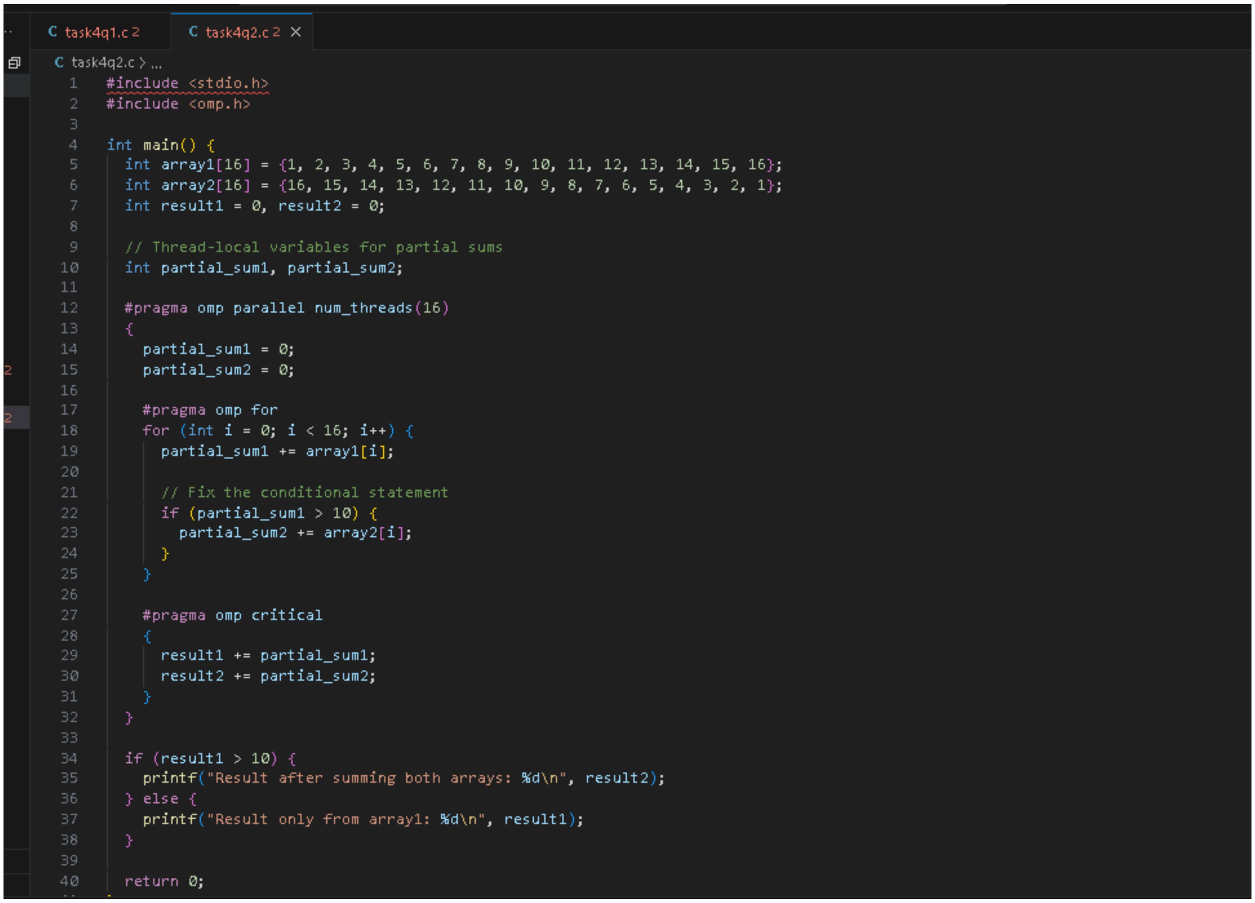
Output:



```
PS C:\Users\mahad\Desktop\pdc_assignments> gcc -o task4q1 -fopenmp task4q1.c
PS C:\Users\mahad\Desktop\pdc_assignments> .\task4q1.exe
272
PS C:\Users\mahad\Desktop\pdc_assignments>
```

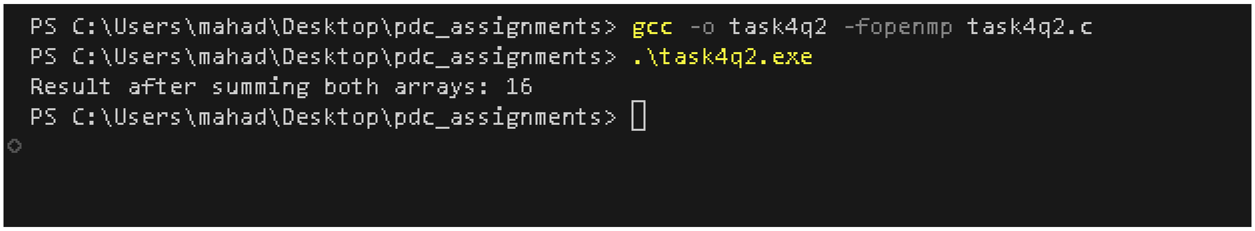
Q2 **Convert it into Parallel using 16 threads.**

Code:



```c
#include <stdio.h>
#include <omp.h>

int main() {
  int array1[16] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16};
  int array2[16] = {16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1};
  int result1 = 0, result2 = 0;

  // Thread-local variables for partial sums
  int partial_sum1, partial_sum2;

  #pragma omp parallel num_threads(16)
  {
    partial_sum1 = 0;
    partial_sum2 = 0;

    #pragma omp for
    for (int i = 0; i < 16; i++) {
      partial_sum1 += array1[i];

      // Fix the conditional statement
      if (partial_sum1 > 10) {
        partial_sum2 += array2[i];
      }
    }

    #pragma omp critical
    {
      result1 += partial_sum1;
      result2 += partial_sum2;
    }
  }

  if (result1 > 10) {
    printf("Result after summing both arrays: %d\n", result2);
  } else {
    printf("Result only from array1: %d\n", result1);
  }

  return 0;
```

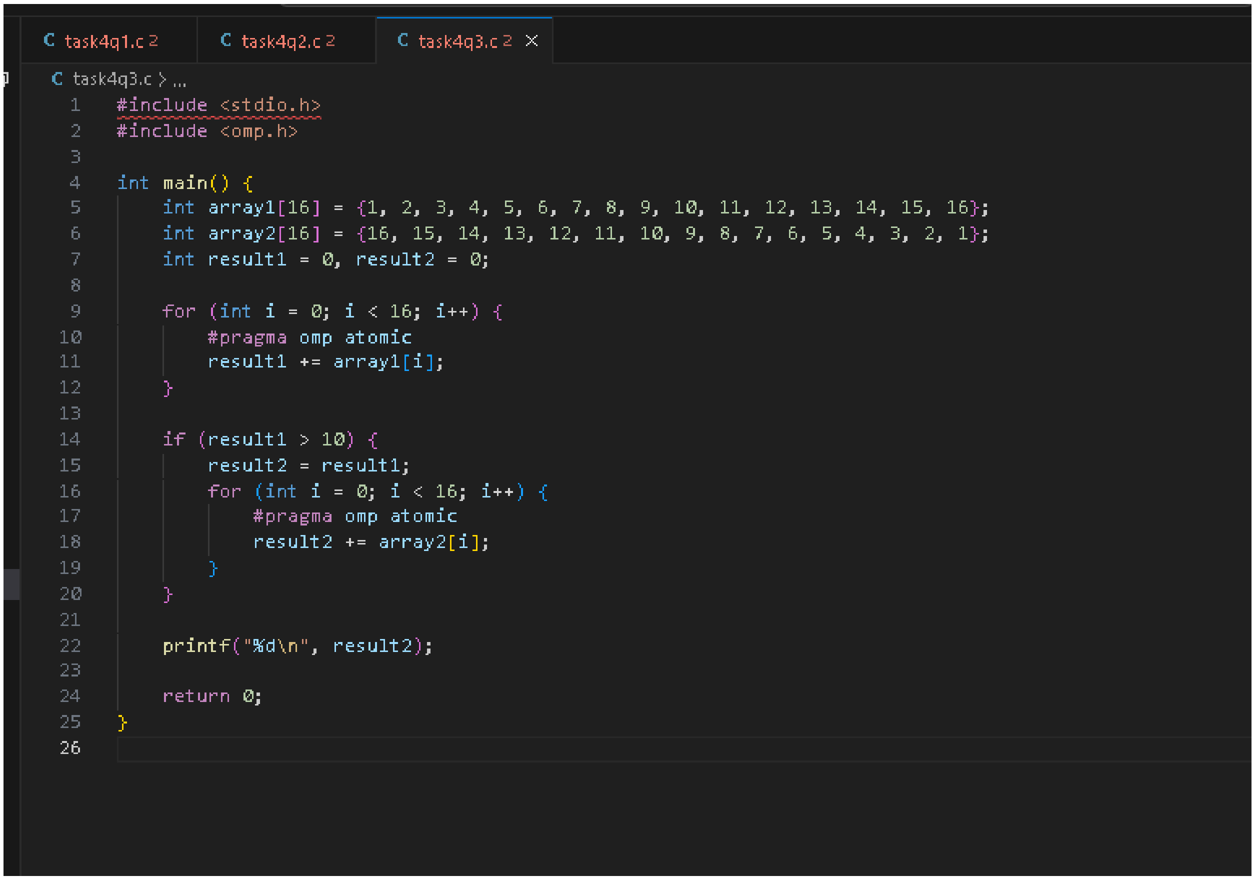
Output:



```
PS C:\Users\mahad\Desktop\pdc_assignments> gcc -o task4q2 -fopenmp task4q2.c
PS C:\Users\mahad\Desktop\pdc_assignments> .\task4q2.exe
Result after summing both arrays: 16
PS C:\Users\mahad\Desktop\pdc_assignments> 
```

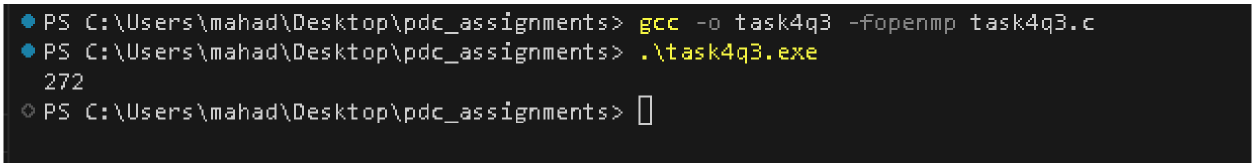
Q3 **Try removing the reduction() clause and add #pragma omp atomic just beore the +=. What is the effect on result? Explain.**

Code :



```c
#include <stdio.h>
#include <omp.h>

int main() {
    int array1[16] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16};
    int array2[16] = {16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1};
    int result1 = 0, result2 = 0;

    for (int i = 0; i < 16; i++) {
        #pragma omp atomic
        result1 += array1[i];
    }

    if (result1 > 10) {
        result2 = result1;
        for (int i = 0; i < 16; i++) {
            #pragma omp atomic
            result2 += array2[i];
        }
    }

    printf("%d\n", result2);

    return 0;
}

```

Output:

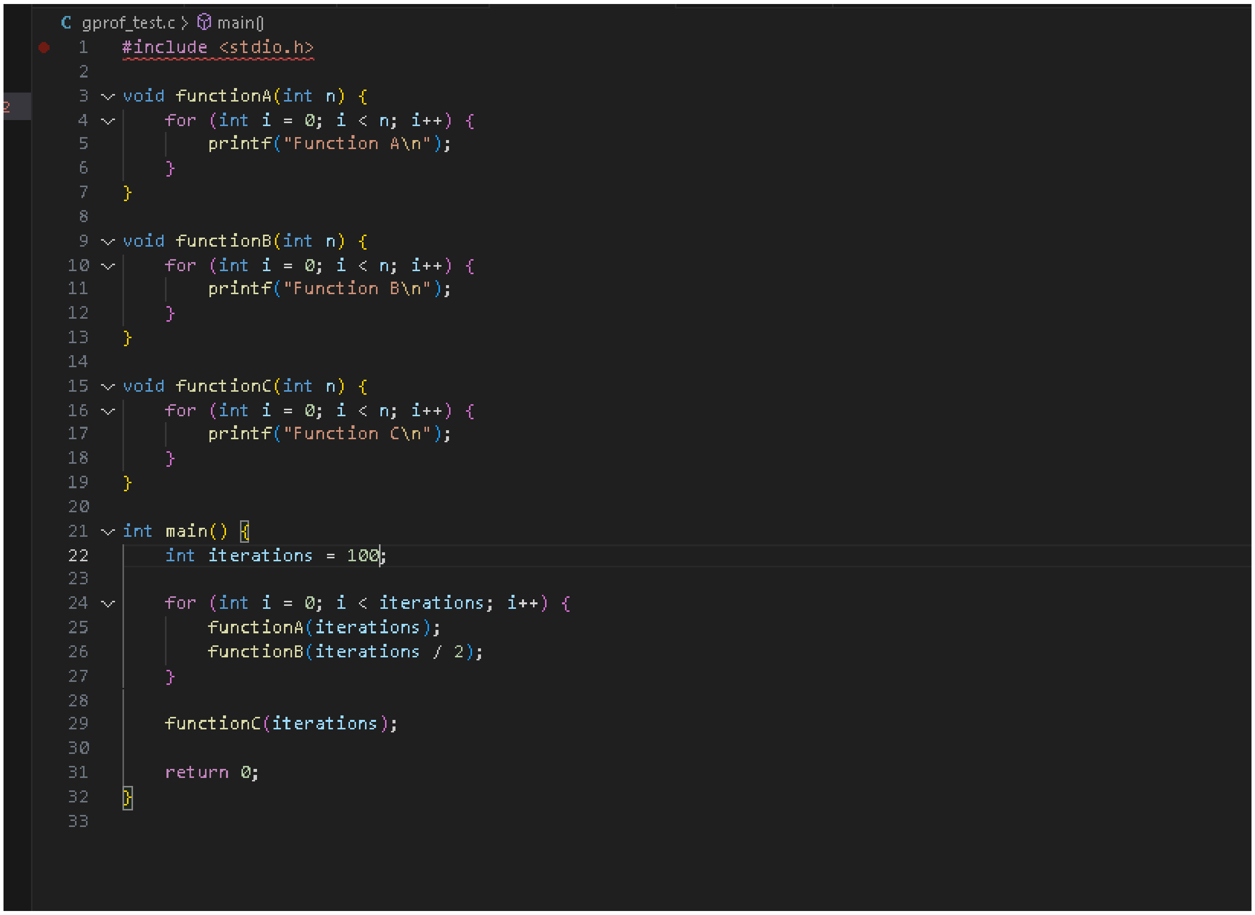


```
PS C:\Users\mahad\Desktop\pdc_assignments> gcc -o task4q3 -fopenmp task4q3.c
PS C:\Users\mahad\Desktop\pdc_assignments> .\task4q3.exe
272
PS C:\Users\mahad\Desktop\pdc_assignments> 
```

Effect of result: The #pragma omp atomic directive ensures that the specified operation is executed atomically, avoiding race conditions that may occur in parallel regions. However, using atomic operations can introduce contention, and in some cases, it might lead to decreased performance compared to using a reduction clause. In this specific code, since the updates to result1 and result2 are performed atomically, the final result should still be correct. However, the performance characteristics may vary depending on the specifics of the system and workload

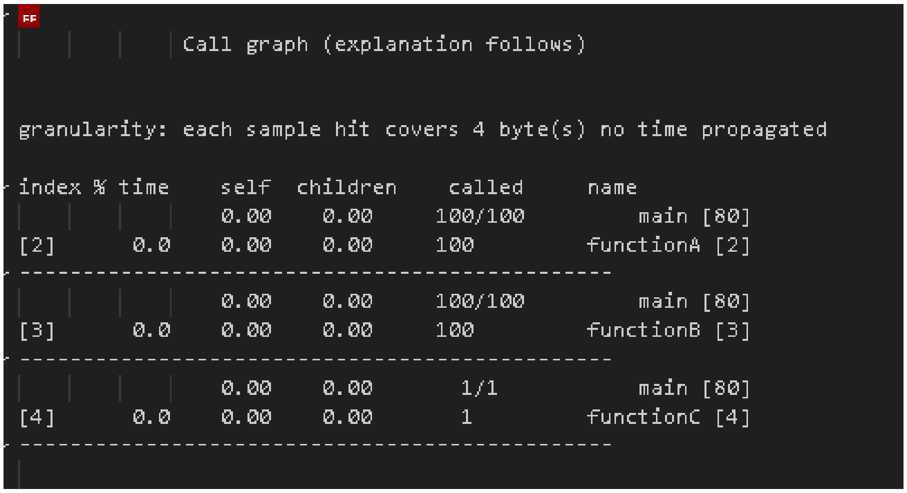
Task 5

Code:

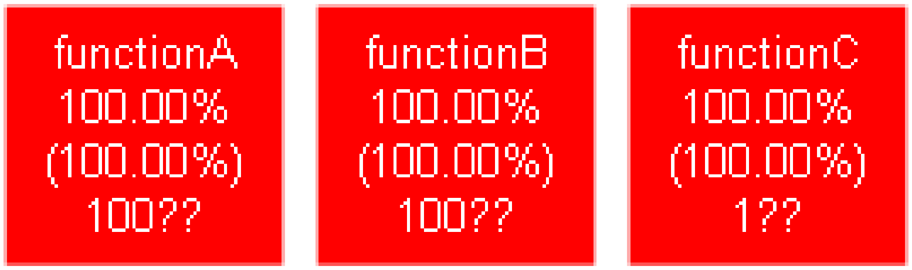


```c
#include <stdio.h>

void functionA(int n) {
    for (int i = 0; i < n; i++) {
        printf("Function A\n");
    }
}

void functionB(int n) {
    for (int i = 0; i < n; i++) {
        printf("Function B\n");
    }
}

void functionC(int n) {
    for (int i = 0; i < n; i++) {
        printf("Function C\n");
    }
}

int main() {
    int iterations = 100;

    for (int i = 0; i < iterations; i++) {
        functionA(iterations);
        functionB(iterations / 2);
    }

    functionC(iterations);

    return 0;
}
```

Output:



```
                 Call graph (explanation follows)


granularity: each sample hit covers 4 byte(s) no time propagated

index % time    self  children    called     name
                0.00    0.00   100/100          main [80]
[2]      0.0    0.00    0.00   100          functionA [2]
-----------------------------------------------
                0.00    0.00   100/100          main [80]
[3]      0.0    0.00    0.00   100          functionB [3]
-----------------------------------------------
                0.00    0.00     1/1            main [80]
[4]      0.0    0.00    0.00     1          functionC [4]
-----------------------------------------------
```


functionA
100.00%
(100.00%)
100??
functionB
100.00%
(100.00%)
100??
functionC
100.00%
(100.00%)
1??