



Creating MapReduce program to calculating pi



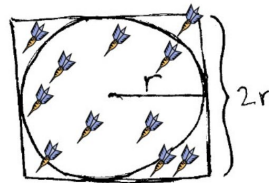
Table of Content

1. Theory
2. Setup
3. Create input files
4. Execution
5. Result
6. Conclusion
7. References

Theory

There are many ways to calculate Pi. But in this project, we are using MapReduce

- Throw N darts on the board. Each dart lands at a random position (x,y) on the board.



- Note if each dart landed inside the circle or not
 - Check if $x^2 + y^2 < r^2$
- Take the total number of darts that landed in the circle as S

$$4 \left(\frac{S}{N} \right) = \pi$$

Formula:

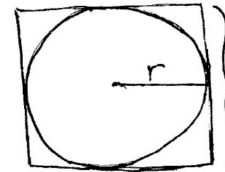
$$4 * S / N = 4 * (\pi * r * r) / (4 * r * r) = \pi$$

Note:

- S = darts inside the circle = the area of the circle
- N = darts on the board = the area of the square

Sample MapReduce Code- Estimate π

- Estimating π by random sampling
- Imagine you have a dart board like so:



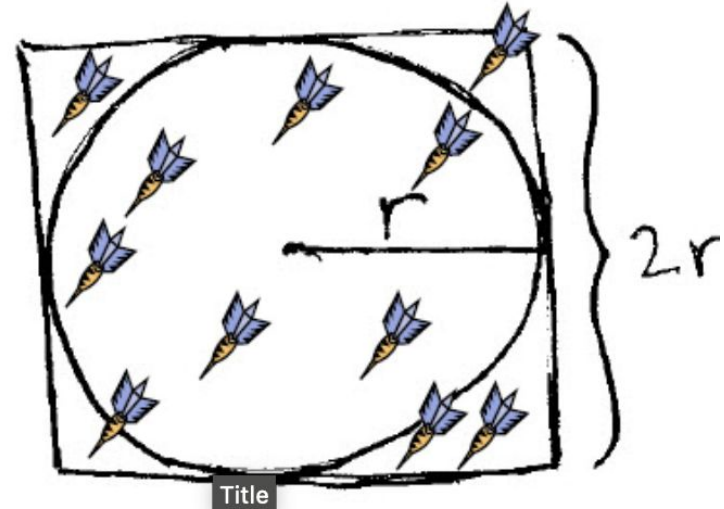
$$P(\text{dart in circle}) = \frac{\pi r^2}{4r^2} = \frac{\pi}{4}$$
$$\Rightarrow \pi = 4P(\text{dart in circle})$$

- π is simply the (ratio of darts that land inside the circle to the total number of darts thrown) times 4

How?

1. Let (x,y) be a random position of the dart inside the square. Then, we map each (x,y) pair to a result. If the pair is inside the circle, then result = 1, otherwise 0.
2. To calculate the π , we need to sum all the pair result inside the circle as S , and divide by the total number of pair N , multiply by 4, and get π .

$$\pi = 4(S/N)$$





Setup

Follow the standard setup for Hadoop in
GCP Ubuntu

```
--2023-06-08 04:24:25-- https://d1cdn.apache.org/hadoop/common/hadoop-3.3.5/hadoop-3.3.5.tar.gz
Resolving d1cdn.apache.org (d1cdn.apache.org)... 151.101.2.132, 2a04:4e42::644
Connecting to d1cdn.apache.org (d1cdn.apache.org)|151.101.2.132|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 706533213 (674M) [application/x-gzip]
Saving to: 'hadoop-3.3.5.tar.gz'

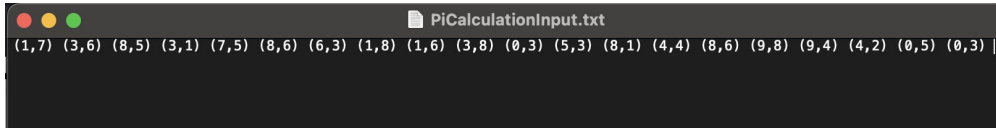
hadoop-3.3.5.tar.gz      100%[=====>] 673.80M  99.9MB/s   in 5.9s

2023-06-08 04:24:53 (114 MB/s) - 'hadoop-3.3.5.tar.gz' saved [706533213/706533213]

jfang757@mapreduce:~$ tar xzf hadoop-3.3.5.tar.gz
jfang757@mapreduce:~$ cd hadoop-3.3.5
jfang757@mapreduce:~/hadoop-3.3.5$ ls -all
total 120
drwxr-xr-x 10 jfang757 jfang757 4096 Mar 15 16:58 .
drwxr-xr-x  5 jfang757 jfang757 4096 Jun  8 04:25 ..
-rw-rw-r-- 1 jfang757 jfang757 24496 Feb 25 09:59 LICENSE-binary
-rw-rw-r-- 1 jfang757 jfang757 15217 Jul 16 2022 LICENSE.txt
-rw-rw-r-- 1 jfang757 jfang757 29473 Jul 16 2022 NOTICE-binary
-rw-rw-r-- 1 jfang757 jfang757 1541 Apr 22 2022 NOTICE.txt
-rw-rw-r-- 1 jfang757 jfang757 175 Apr 22 2022 README.txt
drwxr-xr-x  2 jfang757 jfang757 4096 Mar 15 16:58 bin
drwxr-xr-x  3 jfang757 jfang757 4096 Mar 15 15:58 etc
drwxr-xr-x  2 jfang757 jfang757 4096 Mar 15 16:58 include
drwxr-xr-x  3 jfang757 jfang757 4096 Mar 15 16:58 lib
drwxr-xr-x  4 jfang757 jfang757 4096 Mar 15 16:58 libexec
drwxr-xr-x  2 jfang757 jfang757 4096 Mar 15 16:58 licenses-binary
drwxr-xr-x  3 jfang757 jfang757 4096 Mar 15 15:58 sbin
drwxr-xr-x  4 jfang757 jfang757 4096 Mar 15 17:27 share
jfang757@mapreduce:~/hadoop-3.3.5$ update-alternatives --list java
/usr/lib/jvm/java-8-openjdk-amd64/jre/bin/java
jfang757@mapreduce:~/hadoop-3.3.5$ vi ~/.bashrc
jfang757@mapreduce:~/hadoop-3.3.5$ . ~/.bashrc
jfang757@mapreduce:~/hadoop-3.3.5$ echo $JAVA_HOME
/usr/lib/jvm/java-8-openjdk-amd64
jfang757@mapreduce:~/hadoop-3.3.5$ echo $HADOOP_HOME
/home/jfang757/hadoop-3.3.5
jfang757@mapreduce:~/hadoop-3.3.5$
```

Create input files

Run a java program to create a file contain random number pair.



```
1 import java.io.File;
2 import java.io.FileWriter;
3 import java.io.IOException;
4 import java.util.Scanner;
5 public class GenerateNums{
6     public static void main(String[] args){
7         System.out.println(x;"How many random numbers to generate: ");
8         Scanner input = new Scanner(System. in);
9         int RandomNumCount = input.nextInt();
10
11         System.out.println(x;"What's the radium number? ");
12         int radius = input.nextInt();
13         int diameter = radius * 2;
14         input.close();
15
16         try{
17             File file = new File(pathnames"./PiCalculationInput.txt");
18             file.createNewFile();
19
20             FileWriter writer = new FileWriter(file);
21
22             for(int i = 0; i < RandomNumCount; i++){
23                 int xvalue = (int) (Math.random() * diameter);
24                 int yvalue = (int) (Math.random() * diameter);
25                 writer.write("(" + xvalue + "," + yvalue + ") ");
26                 System.out.print("(" + xvalue + "," + yvalue + ") ");
27             }
28             writer.flush();
29             writer.close();
30         }catch(IOException e){
31             e.printStackTrace();
32         }
33     }
34 }
```

Execution-Pi MapReduce program

src > J PiCalculation.java > PiCalculation > TokenizerMapper > map(Object, Text, Context)

```
1 import java.io.*;
2 import java.util.*;
3 import java.lang.Object;
4 import java.conf.*;
5
6 import org.apache.hadoop.fs.Path;
7 import org.apache.hadoop.conf.*;
8 import org.apache.hadoop.io.*;
9 import org.apache.hadoop.mapreduce.Mapper;
10 import org.apache.hadoop.mapreduce.Mapper.Context;
11 import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
12 import org.apache.hadoop.mapreduce.lib.input.TextInputFormat;
13 import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
14 import org.apache.hadoop.mapreduce.lib.output.TextOutputFormat;
15 import org.apache.hadoop.fs.*;
16
17 public class PiCalculation {
18
19     public static class TokenizerMapper
20         extends Mapper<Object, Text, Text, IntWritable> {
21
22         private final static IntWritable one = new IntWritable(1);
23         private Text word = new Text();
24         private int totalLines = 0;
25
26         public void map(Object key, Text value, Context context) throws IOException, InterruptedException {
27
28             totalLines += 1;
29             String line = value.toString();
30             line = line.replace("target:", replacement);
31             line = line.replace("target:", replacement);
32             line = line.replace("target:", replacement);
33
34             StringTokenizer itr = new StringTokenizer(line);
35             int radius = 5;
36             while (itr.hasMoreTokens()) {
```

src > J PiCalculation.java > PiCalculation > TokenizerMapper > map(Object, Text, Context)

```
33
34     StringTokenizer itr = new StringTokenizer(line);
35     int radius = 5;
36     while (itr.hasMoreTokens()) {
37         String x, y;
38         x = itr.nextToken();
39
40         if (itr.hasMoreTokens()) {
41             y = itr.nextToken();
42         } else {
43             y = "0";
44         }
45
46         int xvalue = (int) (Integer.parseInt(x));
47         int yvalue = (int) (Integer.parseInt(y));
48         double check = Math.sqrt(Math.pow((radius - xvalue), 8) + Math.pow((radius - yvalue), 8));
49
50         if (check < radius) {
51             word.set(strings"inside");
52         } else {
53             word.set(strings"outside");
54         }
55         context.write(word, one);
56     }
57 }
58
59 public static class IntSumReducer
60     extends Reducer<Text, IntWritable, Text, IntWritable> {
61     private IntWritable result = new IntWritable(0);
62     private Context context throws IOException, InterruptedException {
63         for (IntWritable val : values) {
64             sum += val.get();
65         }
66         result.set(sum);
67         context.write(key, result);
68     }
69 }
70
71 }
```

```
71     context.write(key, result);
72 }
73
74 Run | Debug
75 public static void main(String[] args) throws Exception {
76     Configuration conf = new Configuration();
77     Job job = Job.getInstance(conf, "PiCalculation");
78     job.setJarByClass(PiCalculation.class);
79     job.setMapperClass(TokenizerMapper.class);
80     job.setCombinerClass(IntSumReducer.class);
81     job.setReducerClass(IntSumReducer.class);
82     job.setOutputKeyClass(Text.class);
83     job.setOutputValueClass(IntWritable.class);
84     FileInputFormat.addInputPath(job, new Path(args[0]));
85     FileOutputFormat.setOutputPath(job, new Path(args[1]));
86     // System.exit(job.waitForCompletion(true) ? 0 : 1);
87     job.waitForCompletion(true);
88     String filePath = args[1] + "/" + "part-r-000000";
89     Path path = new Path(filePath);
90     FileSystem fs = FileSystem.get(path.toUri(), conf);
91
92     BufferedReader br = new BufferedReader(new InputStreamReader(fs.open(path)));
93
94     String line1, line2;
95
96     String line1, line2;
97
98     line1 = br.readLine();
99     System.out.println(line1);
100     line2 = br.readLine();
101     System.out.println(line2);
102
103     line1 = line1.replace("target:"inside", replacement);
104     line2 = line2.replace("target:"outside", replacement);
105     System.out.println("Inside: " + line1 + ", Outside: " + line2);
106
107     if (line1 != null && line2 != null) {
108         double invalue = Double.valueOf(line1);
109         double outvalue = Double.valueOf(line2);
110         double pi = 4 * (invalue / (invalue + outvalue));
111         System.out.println("Pi: " + pi);
112     }
113
114     fs.close();
115
116 }
117 }
```

Execution

Make the HDFS directories required to execute MapReduce jobs

Copy the input files into the distributed file system

Move .class files to hadoop-3.3.5 directory and create jar

```
6322 NameNode
6691 SecondaryNameNode
6822 Jps
6473 DataNode
jfang757@mapreduce:~/hadoop-3.3.5$ wget http://localhost:9870/
--2023-06-08 04:41:19-- http://localhost:9870/
Resolving localhost (localhost)... 127.0.0.1
Connecting to localhost (localhost)|127.0.0.1|:9870... connected.
HTTP request sent, awaiting response... 302 Found
Location: http://localhost:9870/index.html [following]
--2023-06-08 04:41:19-- http://localhost:9870/index.html
Reusing existing connection to localhost:9870.
HTTP request sent, awaiting response... 200 OK
Length: 1079 (1.1K) [text/html]
Saving to: 'index.html'

index.html                               100%[=====>] 1.05K --.-KB/s  in 0s

2023-06-08 04:41:19 (89.6 MB/s) - 'index.html' saved [1079/1079]

jfang757@mapreduce:~/hadoop-3.3.5$ bin/hdfs dfs -mkdir /user
jfang757@mapreduce:~/hadoop-3.3.5$ bin/hdfs dfs -mkdir /user/jfang757
jfang757@mapreduce:~/hadoop-3.3.5$ bin/hdfs dfs -mkdir /user/jfang757/picalculation
jfang757@mapreduce:~/hadoop-3.3.5$ bin/hdfs dfs -mkdir /user/jfang757/picalculation/input
jfang757@mapreduce:~/hadoop-3.3.5$ bin/hdfs dfs -put ../PiCalculation/input/* /user/jfang757/picalculation/
input
jfang757@mapreduce:~/hadoop-3.3.5$ bin/hdfs dfs -ls /user/jfang757/picalculation/input
Found 1 items
-rw-r--r-- 1 jfang757 supergroup 121 2023-06-08 04:43 /user/jfang757/picalculation/input/file01
jfang757@mapreduce:~/hadoop-3.3.5$ bin/hadoop com.sun.tools.javac.Main ../PiCalculation/PiCalculation.java
jfang757@mapreduce:~/hadoop-3.3.5$ cp ../PiCalculation/*.class .
jfang757@mapreduce:~/hadoop-3.3.5$ cp ../PiCalculation/*.java .
jfang757@mapreduce:~/hadoop-3.3.5$ jar cf wc.jar PiCalculation*.class
jfang757@mapreduce:~/hadoop-3.3.5$ ls
LICENSE-binary  'PiCalculation$IntSumReducer.class'  README.txt  index.html  licenses-binary  share
LICENSE.txt     'PiCalculation$TokenizerMapper.class' bin          input       logs          wc.jar
NOTICE-binary   PiCalculation.class                  etc          lib         output
NOTICE.txt      PiCalculation.java                  include      libexec     sbin
jfang757@mapreduce:~/hadoop-3.3.5$
```




Result

I use 20 pairs with a radius of 5 for this project, and the results is

Inside 14

Outside 6

Pi 2.8

```
...
HDFS: Number of bytes written=20
HDFS: Number of read operations=15
HDFS: Number of large read operations=0
HDFS: Number of write operations=4
HDFS: Number of bytes read erasure-coded=0
Map-Reduce Framework
  Map input records=1
  Map output records=20
  Map output bytes=226
  Map output materialized bytes=33
  Input split bytes=127
  Combine input records=20
  Combine output records=2
  Reduce input groups=2
  Reduce shuffle bytes=33
  Reduce input records=2
  Reduce output records=2
  Spilled Records=4
  Shuffled Maps =1
  Failed Shuffles=0
  Merged Map outputs=1
  GC time elapsed (ms)=66
  Total committed heap usage (bytes)=246947840
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=121
File Output Format Counters
  Bytes Written=20
inside 14
outside 6
Inside:14, Outside:6
PI:2.8
jfang757@mapreduce:~/hadoop-3.3.5$
```



Conclusion

The result 2.8 is far off π , but I only use 20 pairs of numbers. If we increase the number of pairs to 200 or more, the result will be much closer to the π .



References

Exercises for Pi: https://hc.labnet.sfbu.edu/~henry/npu/classes/mapreduce/pi/slide/exercise_pi.html

Sample code:

https://hc.labnet.sfbu.edu/~henry/npu/classes/mapreduce/pi/hw/q1/2022_fall/PiCalculation.html

MapRedcue Pi concept:

https://hc.labnet.sfbu.edu/~henry/npu/classes/mapreduce/pi/slide/mapreduce_pi.html