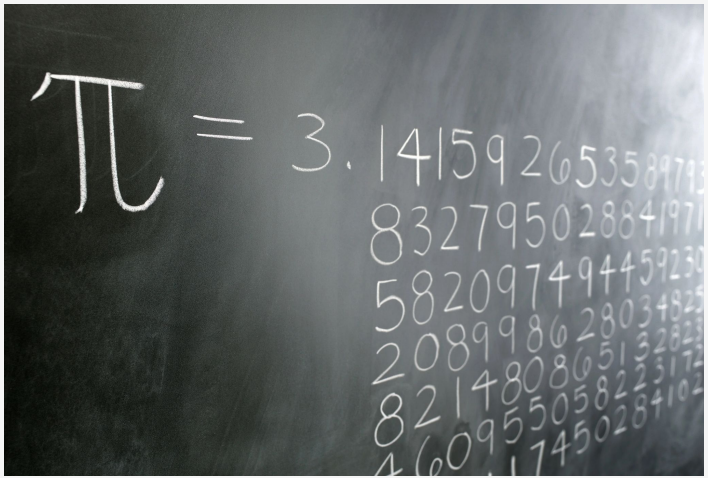
Project: Creating MapReduce program to calculating Pi



CS570 Big Data Processing Project
By Feven Araya
Instructor: Dr. Chang, Henry

Table of contents

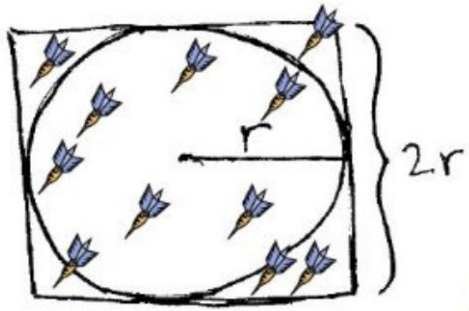
01

Introduction

This Pi Project is to use Google Cloud Platform to implement Hadoop with MapReduce to calculate pi value.

THEORY OF Pi Calculation

- 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 $x^2+y^2<r$
  - 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

The value of pi can be calculated by counting the number of random darts that falls in the circle and outside the circle

02 Design

This section will discuss about the process and methods designed to solve pi calculation.

Technology used

- Using GCP Ubuntu as project environment.
- Using Hadoop framework to implement MapReduce model.
- Program in Java Language.

Job: Pi										
Map Task								Reduce Task		
map()				combine()				reduce()		
Input (Given)		Output (Program)		Input (Given)		Output (Program)		Input (Given)		Output (Program)
Key	Value (radius=2)	Key	Value (radius=2)	Key	Values	Key	Value	Key	Values	
file1	(0, 1)	Outside	1	Inside	[1]	Inside	1	Inside	[1, 3, 1]	Inside 5
	(1, 3)	Inside	1	Outside	[1, 1]	Outside	2	Outside	[2, 1, 4]	Outside 7
	(4, 3)	Outside	1							
file2	(2, 3)	Inside	1	Inside	[1, 1, 1]	Inside	3			
	(1, 3)	Inside	1	Outside	[1]	Outside	1			
	(1, 4)	Outside	1							
	(3, 2)	Inside	1							
file3	(3, 0)	Outside	1	Inside	[1]	Inside	1			
	(3, 3)	Inside	1	Outside	[1, 1, 1, 1]	Outside	4			
	(3, 4)	Outside	1							
	(0, 0)	Outside	1							
	(4, 4)	Outside	1							

Processes

01 Prepare Input File

Write a Java program to generate numbers of random pairs of point(x, y) with given radius - Save the result in file to use as MapReduce input file

02 Code for MapReduce

Write MapReduce program in Java Language to count number of points inside and outside of the circle with given radius.

03 Run Mapreduce on GCP

Using the input file generated in step 1 to run MapReduce program in Step 2
Output should be like: Inside xxx Outside xxx

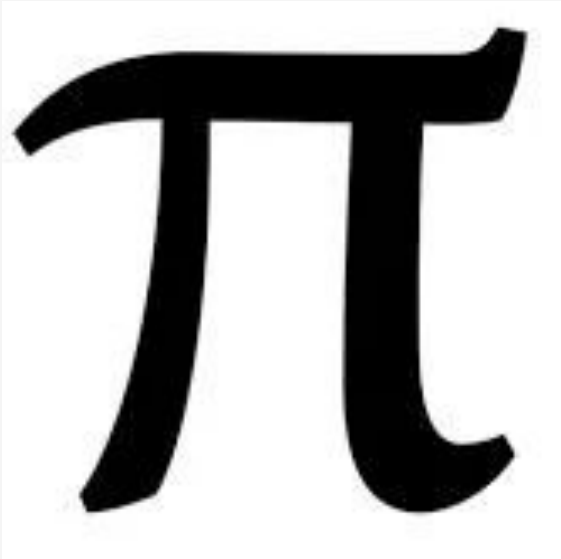
04 Calculate Pi

Write a Java Program to calculate pi value - Using the output from Step 3 get pi value

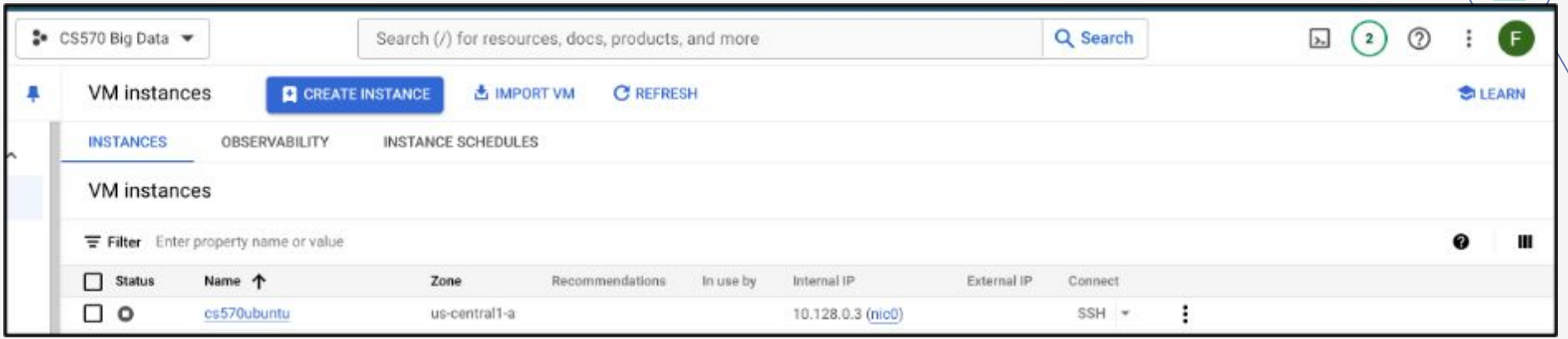
03

Implementation

Getting ready to test



ENVIRONMENT—-GCP



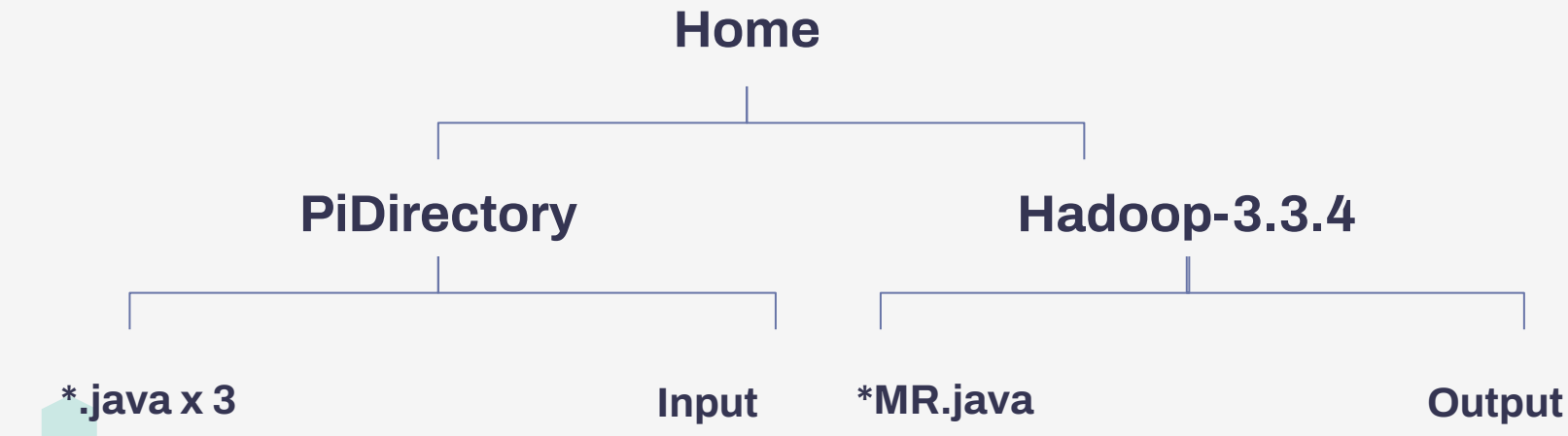
Start VM instance on GCP

ENVIRONMENT—-Connection

```
faraya85431@cs570ubuntu:~$ ssh localhost
The authenticity of host 'localhost (127.0.0.1)' can't be established.
ECDSA key fingerprint is SHA256:Ch5V0vny0H8scvEz/UdxM15ueJH/L0sN1mbqQTzmCF0.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'localhost' (ECDSA) to the list of known hosts.
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.15.0-1060-gcp x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

 System information as of Wed Jun  5 00:17:22 UTC 2024

  System load:  0.38              Processes:             107
  Usage of /:   19.1% of 9.51GB   Users logged in:       0
  Memory usage: 5%                IPv4 address for ens4: 10.128.0.3
  Swap usage:   0%


Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


The list of available updates is more than a week old.
To check for new updates run: sudo apt update
New release '22.04.3 LTS' available.
Run 'do-release-upgrade' to upgrade to it.


Last login: Wed Jun  5 00:17:23 2024 from 35.235.241.16
faraya85431@cs570ubuntu:~$
```

Connect with localhost

Code structure



Create Directory- PiProject

```
faraya85431@cs570ubuntu:~$ ls
PiProject  WordCount  hadoop-3.3.4  hadoop-3.3.4.tar.gz
```

This directory is used to enter necessary java codes.

GenerateDots.java

```java
import java.io.IOException;
import java.util.Random;

public class GenerateDots {
    public static void main(String[] args) throws Exception {
        //args[0]=>radius args[1]=>pairs of (x,y) to create
        //convert arguments to integer
        double radius = Double.parseDouble(args[0]);
        int num = Integer.parseInt(args[1]);
        for (int i=0; i< num; i++){
            double x = Math.random()*2*radius;
            double y = Math.random()*2*radius;

            System.out.println( Double.toString(x) + ' ' + Double.toString(y) + ' ' + Double.toString(radius));
        }
    }
}
```

Create java file called *GenerateDots.java* to generate random dot pairs with command line arguments taken in as radius and number of pairs. Output format: x y radius

CalculatePiMR.java

```java
import java.io.IOException; import java.util.*;
import java.lang.Object;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.conf.*;
import org.apache.hadoop.io.*;
import org.apache.hadoop.mapreduce.*;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.input.TextInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import org.apache.hadoop.mapreduce.lib.output.TextOutputFormat;

public class CalculatePiMR {
   public static class Map extends Mapper<LongWritable, Text, Text,IntWritable>
   {
        private final static IntWritable one = new IntWritable(1);
        private Text word = new Text();

        public void map(LongWritable key, Text value, Context context) throws IOException, InterruptedException
        {
            String line = value.toString();
            StringTokenizer tokenizer = new StringTokenizer(line);

            while(tokenizer.hasMoreTokens()){
                String xStr="0", yStr="0", rStr="0";
                xStr = tokenizer.nextToken();
                if(tokenizer.hasMoreTokens()){
                        yStr = tokenizer.nextToken();
                }
                if(tokenizer.hasMoreTokens()){
                        rStr = tokenizer.nextToken();
                }

                Double x = (Double)(Double.parseDouble(xStr));
                Double y = (Double)(Double.parseDouble(yStr));
                Double r = (Double)(Double.parseDouble(rStr));

                Double check = Math.pow(x-r, 2) + Math.pow(y-r, 2) - Math.pow(r, 2);
                if(check <= 0){
                        word.set("Inside");
                }else{
                        word.set("Outside");
                }
                context.write(word, one);
```

```java
                }else{
                        word.set("Outside");
                }
                context.write(word, one);
            }

        }
    }

  public static class Reduce extends Reducer<Text, IntWritable,Text, IntWritable>
    {
        public void reduce(Text key, Iterable<IntWritable> values,Context context) throws IOException, InterruptedException
        {
            int sum = 0;
            for (IntWritable val : values) {
                sum += val.get();
            }
            context.write(key, new IntWritable(sum));
        }
    }

public static void main(String[] args) throws Exception
{
      Configuration conf = new Configuration();

      Job job = new Job(conf, "CalculatePiMR");
      job.setJarByClass(CalculatePiMR.class);
      job.setOutputKeyClass(Text.class);
      job.setOutputValueClass(IntWritable.class);

      job.setMapperClass(Map.class);
      job.setReducerClass(Reduce.class);

      job.setInputFormatClass(TextInputFormat.class);
      job.setOutputFormatClass(TextOutputFormat.class);

      FileInputFormat.addInputPath(job, new Path(args[0]));
      FileOutputFormat.setOutputPath(job, new Path(args[1]));

      job.waitForCompletion(true);
   }
}
```

create CalculatePiMR.java java file that reads the results of a MapReduce job from a file and calculates the value of Pi based on counts of points inside and outside a unit circle obtained from the file's last two lines.

CalculatePi.java

```
U nano 4.8                                                    CalculatePi.java
rt java.io.*;
ic class CalculatePi {
    public static void main(String[] args) throws Exception{
            String file = "../hadoop-3.3.4/"+args[0]+"/part-r-00000";
            BufferedReader bufferedReader = new BufferedReader(new FileReader(file));

            String curLine="", line1="", line2="";
            while ((curLine = bufferedReader.readLine()) != null){
                    line1 = curLine;
                    if((curLine = bufferedReader.readLine()) != null){
                            line2 = curLine;
                    }
            }
            System.out.println(line1);
            System.out.println(line2);

            //System.out.println(line1.length() + " " + line2.length());
            String in = line1.substring(line1.length()-(line1.length()-6-1));
            String out = line2.substring(line2.length()-(line2.length()-7-1));

            double inside = Double.parseDouble(in);
            //System.out.println(inside);
            double outside = Double.parseDouble(out);
            //System.out.println(outside);
            double pi = 4 * ( inside / ( inside + outside ) );
            System.out.println("PI value is: " + pi );

            bufferedReader.close();
    }
```

Create java file called *CalculatePi.java* to show Hadoop MapReduce program to calculate Pi using the Monte Carlo method, consisting of a mapper that calculates whether points fall inside or outside a unit circle and a reducer that sums these counts to estimate Pi.

Code Structure

```
faraya85431@cs570ubuntu:~/PiProject$ ls
CalculatePi.java  CalculatePiMR.java  GenerateDots.java
```

04

Test

Process to test the project

Steps

```
faraya85431@cs570-ubuntu:~/hadoop-3.3.4$ bin/hdfs namenode -format
WARNING: /home/faraya85431/hadoop-3.3.4/logs does not exist. Creating.
2024-05-28 03:34:18,386 INFO namenode.NameNode: STARTUP_MSG:
/************************************************************
STARTUP_MSG: Starting NameNode
STARTUP_MSG:   host = cs570-ubuntu.us-central1-a.c.cs570-big-data-424622.internal/10.128.0.2
STARTUP_MSG:   args = [-format]
STARTUP_MSG:   version = 3.3.4
STARTUP_MSG:   classpath = /home/faraya85431/hadoop-3.3.4/etc/hadoop:/home/faraya85431/hadoop-3.3.4/share/hadoop/common/lib/hadoop-annotations-3.3.4.jar:/home/faraya85431/hadoop-3.
3.4/share/hadoop/common/lib/netty-3.10.6.Final.jar:/home/faraya85431/hadoop-3.3.4/share/hadoop/common/lib/jackson-databind-2.12.7.jar:/home/faraya85431/hadoop-3.3.4/share/hadoop/co
mmon/lib/jakarta.activation-api-1.2.1.jar:/home/faraya85431/hadoop-3.3.4/share/hadoop/common/lib/curator-framework-4.2.0.jar:/home/faraya85431/hadoop-3.3.4/share/hadoop/common/lib/
jetty-security-9.4.43.v20210629.jar:/home/faraya85431/hadoop-3.3.4/share/hadoop/common/lib/kerby-util-1.0.1.jar:/home/faraya85431/hadoop-3.3.4/share/hadoop/common/lib/protobuf-java
-2.5.0.jar:/home/faraya85431/hadoop-3.3.4/share/hadoop/common/lib/jaxb-api-2.2.11.jar:/home/faraya85431/hadoop-3.3.4/share/hadoop/common/lib/jackson-jaxrs-1.9.13.jar:/home/faraya85
431/hadoop-3.3.4/share/hadoop/common/lib/jackson-mapper-asl-1.9.13.jar:/home/faraya85431/hadoop-3.3.4/share/hadoop/common/lib/jetty-webapp-9.4.43.v20210629.jar:/home/faraya85431/ha
doop-3.3.4/share/hadoop/common/lib/kerby-asn1-1.0.1.jar:/home/faraya85431/hadoop-3.3.4/share/hadoop/common/lib/jetty-http-9.4.43.v20210629.jar:/home/faraya85431/hadoop-3.3.4/share/
hadoop/common/lib/paranamer-2.3.jar:/home/faraya85431/hadoop-3.3.4/share/hadoop/common/lib/jetty-io-9.4.43.v20210629.jar:/home/faraya85431/hadoop-3.3.4/share/hadoop/common/lib/snap
py-java-1.1.8.2.jar:/home/faraya85431/hadoop-3.3.4/share/hadoop/common/lib/jackson-annotations-2.12.7.jar:/home/faraya85431/hadoop-3.3.4/share/hadoop/common/lib/commons-lang3-3.12.
0.jar:/home/faraya85431/hadoop-3.3.4/share/hadoop/common/lib/slf4j-api-1.7.36.jar:/home/faraya85431/hadoop-3.3.4/share/hadoop/common/lib/jetty-util-9.4.43.v20210629.jar:/home/faray
a85431/hadoop-3.3.4/share/hadoop/common/lib/animal-sniffer-annotations-1.17.jar:/home/faraya85431/hadoop-3.3.4/share/hadoop/common/lib/re2j-1.1.jar:/home/faraya85431/hadoop-3.3.4/s
hare/hadoop/common/lib/jackson-core-2.12.7.jar:/home/faraya85431/hadoop-3.3.4/share/hadoop/common/lib/jsr305-3.0.2.jar:/home/faraya85431/hadoop-3.3.4/share/hadoop/common/lib/audien
ce-annotations-0.5.0.jar:/home/faraya85431/hadoop-3.3.4/share/hadoop/common/lib/guava-27.0-jre.jar:/home/faraya85431/hadoop-3.3.4/share/hadoop/common/lib/commons-io-2.8.0.jar:/home
/faraya85431/hadoop-3.3.4/share/hadoop/common/lib/kerby-pkix-1.0.1.jar:/home/faraya85431/hadoop-3.3.4/share/hadoop/common/lib/commons-math3-3.1.1.jar:/home/faraya85431/hadoop-3.3.4
```

Format the file system

Steps

```
faraya85431@cs570ubuntu:~$ cd hadoop-3.3.4
faraya85431@cs570ubuntu:~/hadoop-3.3.4$ ls
LICENSE-binary  LICENSE.txt  NOTICE-binary  NOTICE.txt  README.txt  bin  etc  include  input  lib  libexec  licenses-binary  logs  output  sbin  share
faraya85431@cs570ubuntu:~/hadoop-3.3.4$ sbin/start-dfs.sh
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [cs570ubuntu]
```

Start NameNode daemon and DataNode daemon Permission Denied, need to connect ssh again.

Steps

```
faraya85431@ca570ubuntu:~/hadoop-3.3.4$ wget http://localhost:9870/
--2024-06-05 01:22:02--  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]
--2024-06-05 01:22:02--  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

2024-06-05 01:22:02 (136 MB/s) - 'index.html' saved [1079/1079]
```

Test Connection with localhost

Steps

```
faraya85431@cs570ubuntu:~/hadoop-3.3.4$ ls
LICENSE-binary  LICENSE.txt  NOTICE-binary  NOTICE.txt  README.txt  bin  etc  include  index.html  input  lib  libexec  licenses-binary  logs  output  sbin  share
faraya85431@cs570ubuntu:~/hadoop-3.3.4$ cd
faraya85431@cs570ubuntu:~$ ls
PiProject  WordCount  hadoop-3.3.4  hadoop-3.3.4.tar.gz
faraya85431@cs570ubuntu:~$ cd PiProject
faraya85431@cs570ubuntu:~/PiProject$ ls
CalculatePi.java  CalculatePiMR.java  GenerateDots.java  input
faraya85431@cs570ubuntu:~/PiProject$ javac GenerateDots.java
faraya85431@cs570ubuntu:~/PiProject$ ls
CalculatePi.java  CalculatePiMR.java  GenerateDots.class  GenerateDots.java  input
```

```
faraya85431@cs570ubuntu:~/PiProject$ java GenerateDots 5 1000 > ./input/dots.txt
faraya85431@cs570ubuntu:~/PiProject$
```

Compile and run java program to generate dots with radius=5, number = 1000 Output save in ./Input/dots.txt

Steps

```
faraya85431@cs570ubuntu:~/hadoop-3.3.4$ bin/hdfs dfs -mkdir /user/faraya85431
faraya85431@cs570ubuntu:~/hadoop-3.3.4$ bin/hdfs dfs -mkdir /user/faraya85431/PiProject
faraya85431@cs570ubuntu:~/hadoop-3.3.4$ bin/hdfs dfs -mkdir /user/faraya85431/PiProject/input
faraya85431@cs570ubuntu:~/hadoop-3.3.4$
```

```
faraya85431@cs570ubuntu:~/hadoop-3.3.4$ bin/hdfs dfs -put ../PiProject/input/* PiProject/input
```

```
faraya85431@cs570ubuntu:~/hadoop-3.3.4$ bin/hdfs dfs -ls PiProject/input
Found 1 items
-rw-r--r--   1 faraya85431 supergroup      40544 2024-06-05 01:36 PiProject/input/dots.txt
```

Copy file from local to hadoop and check

Steps

```
faraya85431@cs570ubuntu:~/hadoop-3.3.4$ javac -classpath $HADOOP_HOME/share/hadoop/common/hadoop-common-3.3.4.jar:$HADOOP_HOME/share/hadoop/mapreduce/hadoop-mapreduce-client-core-3
.3.4.jar CalculatePiMR.java
Note: CalculatePiMR.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
faraya85431@cs570ubuntu:~/hadoop-3.3.4$ ls
'CalculatePiMR$Map.class'      CalculatePiMR.class   LICENSE-binary   NOTICE-binary   README.txt   etc       index.html   lib       licenses-binary   output   share
'CalculatePiMR$Reduce.class'   CalculatePiMR.java    LICENSE.txt      NOTICE.txt      bin          include   input        libexec   logs              sbin
faraya85431@cs570ubuntu:~/hadoop-3.3.4$
```

Compile Mapreduce program in Hadoop with *.class files created

Steps

```
faraya85431@cs570ubuntu:~/hadoop-3.3.4$ jar cf pi.jar CalculatePiMR*.class
```

Create .jar file with *.class files

Steps

```
faraya85431@cs570ubuntu:~/hadoop-3.3.4$ ./bin/hadoop jar pi.jar CalculatePiMR /user/faraya85431/PiProject/input /user/faraya85431/PiProject/Output

2024-06-05 02:27:54,690 INFO impl.MetricsConfig: Loaded properties from hadoop-metrics2.properties
2024-06-05 02:27:54,812 INFO impl.MetricsSystemImpl: Scheduled Metric snapshot period at 10 second(s).
2024-06-05 02:27:54,812 INFO impl.MetricsSystemImpl: JobTracker metrics system started
2024-06-05 02:27:55,082 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your ap
ner to remedy this.
2024-06-05 02:27:55,285 INFO input.FileInputFormat: Total input files to process : 1
2024-06-05 02:27:55,315 INFO mapreduce.JobSubmitter: number of splits:1
2024-06-05 02:27:55,491 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_local2042586096_0001
2024-06-05 02:27:55,491 INFO mapreduce.JobSubmitter: Executing with tokens: []
2024-06-05 02:27:55,709 INFO mapreduce.Job: The url to track the job: http://localhost:8080/
2024-06-05 02:27:55,710 INFO mapreduce.Job: Running job: job_local2042586096_0001
```

Run MapReduce Program with input file and save result in Output

Steps

```
faraya85431@cs570ubuntu:~/hadoop-3.3.4$ ./bin/hadoop jar pi.jar CalculatePiMR /user/faraya85431/PiProject/input /user/faraya85431/PiProject/Output

2024-06-05 02:27:54,690 INFO impl.MetricsConfig: Loaded properties from hadoop-metrics2.properties
2024-06-05 02:27:54,812 INFO impl.MetricsSystemImpl: Scheduled Metric snapshot period at 10 second(s).
2024-06-05 02:27:54,812 INFO impl.MetricsSystemImpl: JobTracker metrics system started
2024-06-05 02:27:55,082 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and exe
ner to remedy this.
2024-06-05 02:27:55,285 INFO input.FileInputFormat: Total input files to process : 1
2024-06-05 02:27:55,315 INFO mapreduce.JobSubmitter: number of splits:1
2024-06-05 02:27:55,491 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_local2042586096_0001
```

Get output and save to local

RESULT

```
faraya85431@cs570ubuntu:~/hadoop-3.3.4$ bin/hdfs dfs -get PiProject/Output Output
faraya85431@cs570ubuntu:~/hadoop-3.3.4$ cat Output/*
Inside  778
Outside 222
faraya85431@cs570ubuntu:~/hadoop-3.3.4$
```

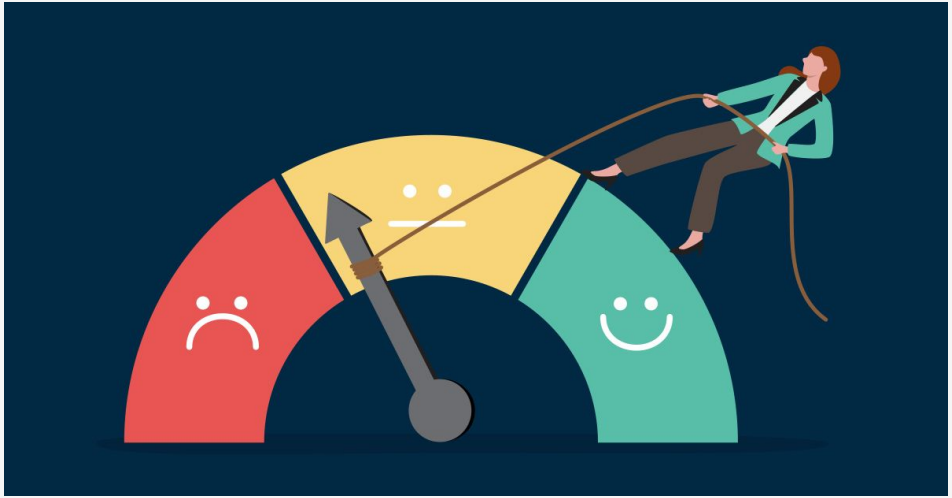
Display Output

RESULT

```
faraya85431@cs570ubuntu:~/PiProject$ ls
CalculatePi.java  CalculatePiMR.java  GenerateDots.java  input
faraya85431@cs570ubuntu:~/PiProject$ javac CalculatePi.java
faraya85431@cs570ubuntu:~/PiProject$ java CalculatePi Output
Inside  778
Outside 222
PI value is: 3.112
faraya85431@cs570ubuntu:~/PiProject$
```

Using the output (local output folder as command line arguments) from MapReduce Program to compile and run java program to get pi value

Pi value calculated is 3.112, and it is almost similar to 3.1415926



05 Enhancements

Can we get better result?

ENHANCED RESULT —- Decrease Radius

```
faraya85431@cs570ubuntu:~/PiProject$ java GenerateDots 1 1000 > ./input/test1.txt
faraya85431@cs570ubuntu:~/PiProject$ ls ./input
dots.txt  test1.txt
faraya85431@cs570ubuntu:~/PiProject$ cat ./input/test1.txt
1.0809598733954442 1.7526935133768917 1.0
1.566208687782557 0.38460898136416444 1.0
0.4936570564384508 1.2099093539919004E-4 1.0
1.2420040202925011 1.0509229713316524 1.0
0.6650333069648484 1.7035116248991937 1.0
1.9134608448293178 0.6993348446066441 1.0
1.5852194903371215 0.6879995588926533 1.0
0.2666664595720678 0.556644795577641 1.0
```

The result can be enhanced by decreasing the radius. Here, radius is 1 and number is 1000.

```
faraya85431@cs570ubuntu:~/hadoop-3.3.4$ bin/hdfs dfs -put ../PiProject/input/test1.txt PiProject/input
faraya85431@cs570ubuntu:~/hadoop-3.3.4$ bin/hdfs dfs -ls PiProject/input
Found 2 items
-rw-r--r--   1 faraya85431 supergroup      40544 2024-06-05 01:36 PiProject/input/dots.txt
-rw-r--r--   1 faraya85431 supergroup      41993 2024-06-05 02:42 PiProject/input/test1.txt
```

```
faraya85431@cs570ubuntu:~/hadoop-3.3.4$ bin/hadoop jar pi.jar CalculatePiMR /user/faraya85431/PiProject/input/test1.txt /user/faraya85431/PiProject/Test1
2024-06-05 02:46:05,637 INFO impl.MetricsConfig: Loaded properties from hadoop-metrics2.properties
2024-06-05 02:46:05,753 INFO impl.MetricsSystemImpl: Scheduled Metric snapshot period at 10 second(s).
2024-06-05 02:46:05,754 INFO impl.MetricsSystemImpl: JobTracker metrics system started
2024-06-05 02:46:06,003 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute you
ner to remedy this.
2024-06-05 02:46:06,143 INFO input.FileInputFormat: Total input files to process : 1
2024-06-05 02:46:06,230 INFO mapreduce.JobSubmitter: number of splits:1
2024-06-05 02:46:06,408 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_local826590554_0001
```

```
faraya85431@cs570ubuntu:~/hadoop-3.3.4$ bin/hdfs dfs -get /user/faraya85431/PiProject/Test1 Test1
faraya85431@cs570ubuntu:~/hadoop-3.3.4$
```

```
faraya85431@cs570ubuntu:~/hadoop-3.3.4$ cat Test1/*
Inside  795
Outside 205
```

```
faraya85431@cs570ubuntu:~/PiProject$ java CalculatePi Test1
Inside  795
Outside 205
PI value is: 3.18
```

Pi value calculate is 3.18 which is also a better value to the real pi value

ENHANCED RESULT —- Increase Number

```
faraya85431@cs570ubuntu:~/PiProject$ java GenerateDots 1 1000 > ./input/test1.txt
faraya85431@cs570ubuntu:~/PiProject$ ls ./input
dots.txt  test1.txt
faraya85431@cs570ubuntu:~/PiProject$ cat ./input/test1.txt
1.0809598733954442 1.7526935133768917 1.0
1.566208687782557 0.38460898136416444 1.0
0.4936570564384508 1.2099093539919004E-4 1.0
1.2420040202925011 1.0509229713316524 1.0
0.6650333069648484 1.7035116248991937 1.0
1.9134608448293178 0.6993348446066441 1.0
1.5852194903371215 0.6879995588926533 1.0
0.2666664595720678 0.556644795577641 1.0
```

The result can also be enhanced by decreasing the radius. Here, radius is 1 and number is 1000.

```
faraya85431@cs570ubuntu:~/hadoop-3.3.4$ bin/hdfs dfs -put ../PiProject/input/test2.txt PiProject/input
faraya85431@cs570ubuntu:~/hadoop-3.3.4$ bin/hdfs dfs -ls PiProject/input
Found 3 items
-rw-r--r--   1 faraya85431 supergroup      40544 2024-06-05 01:36 PiProject/input/dots.txt
-rw-r--r--   1 faraya85431 supergroup      41993 2024-06-05 02:42 PiProject/input/test1.txt
-rw-r--r--   1 faraya85431 supergroup   40537926 2024-06-05 03:18 PiProject/input/test2.txt
faraya85431@cs570ubuntu:~/hadoop-3.3.4$
```

```
faraya85431@cs570ubuntu:~/hadoop-3.3.4$  bin/hadoop jar pi.jar CalculatePiMR /user/faraya85431/PiProject/input/test2.txt /user/faraya85431/PiProject/Test2
2024-06-05 03:49:39,682 INFO impl.MetricsConfig: Loaded properties from hadoop-metrics2.properties
2024-06-05 03:49:39,795 INFO impl.MetricsSystemImpl: Scheduled Metric snapshot period at 10 second(s).
2024-06-05 03:49:39,795 INFO impl.MetricsSystemImpl: JobTracker metrics system started
2024-06-05 03:49:40,070 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your
ner to remedy this.
2024-06-05 03:49:40,202 INFO input.FileInputFormat: Total input files to process : 1
2024-06-05 03:49:40,293 INFO mapreduce.JobSubmitter: number of splits:1
2024-06-05 03:49:40,474 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_local1818677410_0001
2024-06-05 03:49:40,474 INFO mapreduce.JobSubmitter: Executing with tokens: []
2024-06-05 03:49:40,744 INFO mapreduce.Job: The url to track the job: http://localhost:8080/
```

```
faraya85431@cs570ubuntu:~/hadoop-3.3.4$ cat Test2/*
Inside  784866
Outside 215134
```

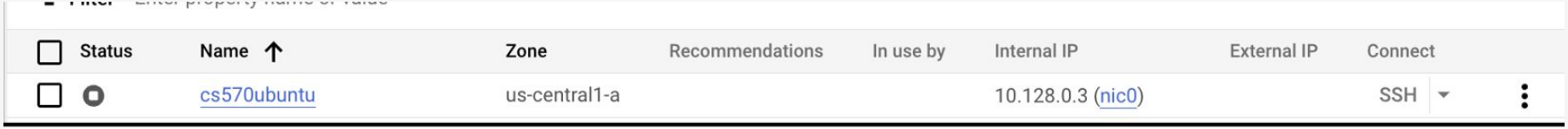
RESULT

```
faraya85431@cs570ubuntu:~/PiProject$ java CalculatePi Test2
Inside  784866
Outside 215134
PI value is: 3.139464
faraya85431@cs570ubuntu:~/PiProject$
```

Pi value calculate is 3.139464 which is very close to the real pi value.

Stop Instance on GCP

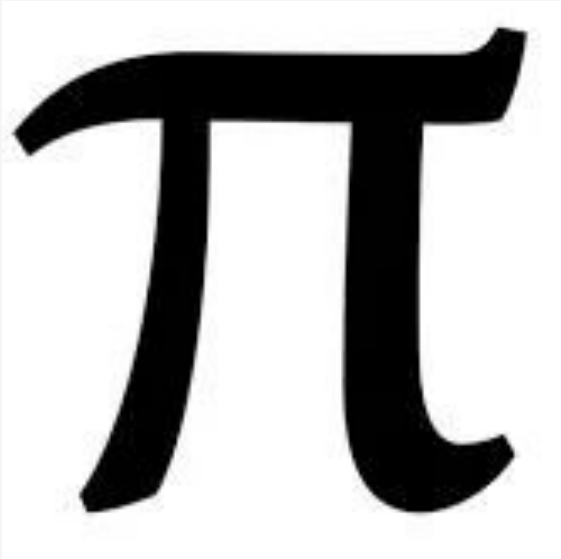
```
faraya85431@cs570-ubuntu:~/hadoop-3.3.4$ sbin/stop-dfs.sh
Stopping namenodes on [localhost]
Stopping datanodes
Stopping secondary namenodes [cs570-ubuntu]
```



After done with project, stop namenode and stop the instance on GCP.

06 Conclusion

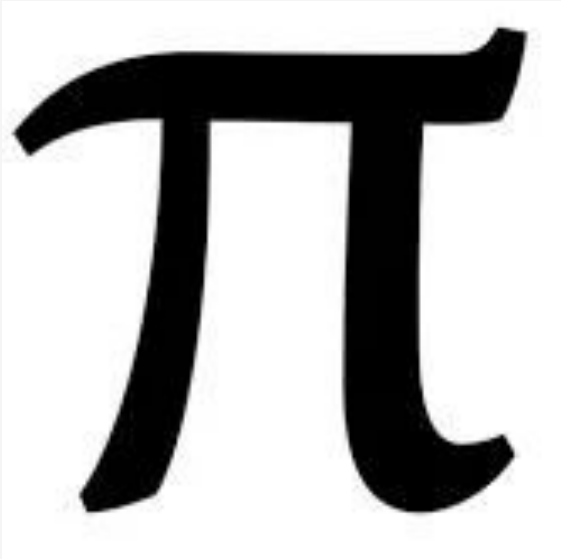
Summarize for Pi Project



- Increased Random Sampling: Generating more random dots enhances the accuracy of the estimated pi value, influenced by the circle's radius and the total number of dots.

- Efficiency of MapReduce: MapReduce excels at processing large datasets quickly and efficiently, utilizing minimal memory.

07
References



A Hadoop application to calculate Pi

Yarn MapReduce approximate-pi example fails exit code 1 when run as non-hadoop user

What is MapReduce in Hadoop? Big Data Architecture.

Thanks!