PI PROJECT

CS570 Big Data Processing Project

By Lidia Bereketeab

TABLE OF CONTENTS

INTRODUCTION

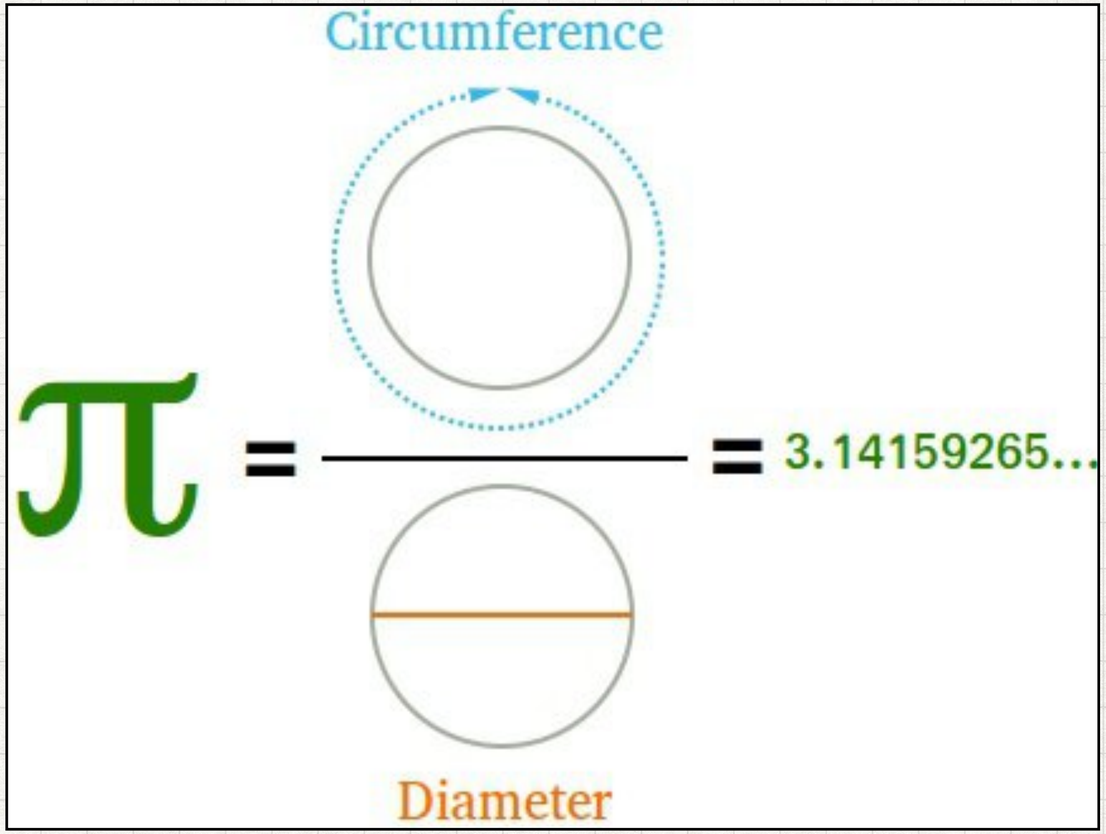
In this project, we leverage Google Cloud Platform to deploy Hadoop with MapReduce for calculating the value of Pi.

This approach showcases the power of cloud computing and distributed processing in handling complex computations efficiently.

THEORY OF PI CALCULATION

Pi Calculation: A Simple Definition

Pi calculation refers to the process of determining the value of Pi (π), a mathematical constant representing the ratio of a circle's circumference to its diameter. This value is approximately 3.14159 and is used in various mathematical and scientific calculations.



THEORY OF PI CALCULATION

Monte Carlo Method for Pi Estimation

- Use a square dartboard with an inscribed circle.
- Randomly throw darts at the dartboard.

THEORY OF PI CALCULATION

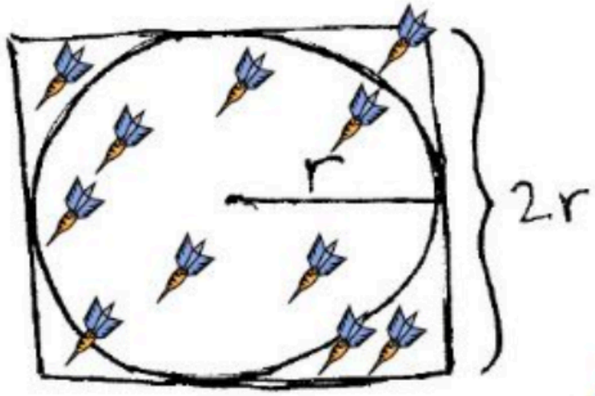
Monte Carlo Method for Pi Estimation : Counting Darts

- **Inside the Circle:** Number of darts that land within the circle.
- **Inside the Square but Outside the Circle:** Number of darts that land within the square but outside the circle.

Approximating Pi:

- The ratio of darts inside the circle to the total number of darts approximates Pi.
- This method leverages statistical randomness and area ratios.

- 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

THEORY OF PI CALCULATION

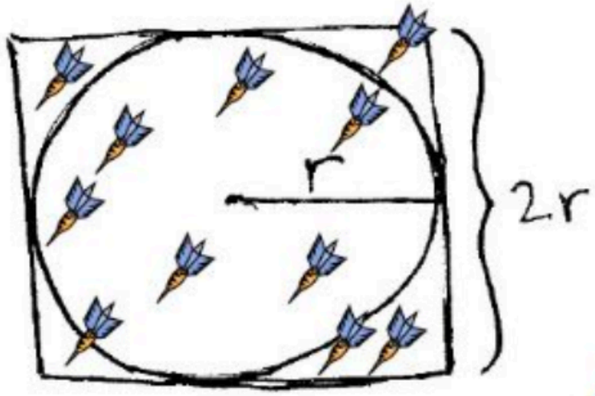
Monte Carlo Method for Pi Estimation : Counting Darts

- **Inside the Circle:** Number of darts that land within the circle.
- **Inside the Square but Outside the Circle:** Number of darts that land within the square but outside the circle.

Approximating Pi:

- The ratio of darts inside the circle to the total number of darts approximates Pi.
- This method leverages statistical randomness and area ratios.

- 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



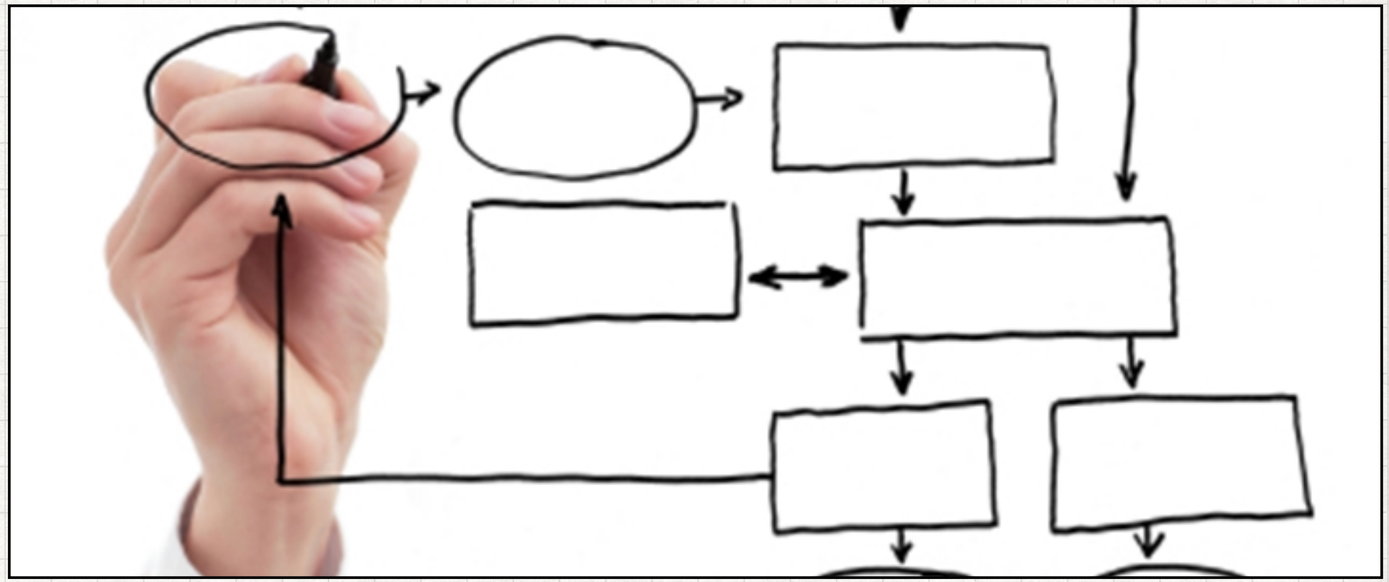
$$4\left(\frac{S}{N}\right)=\pi$$

Formula:

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

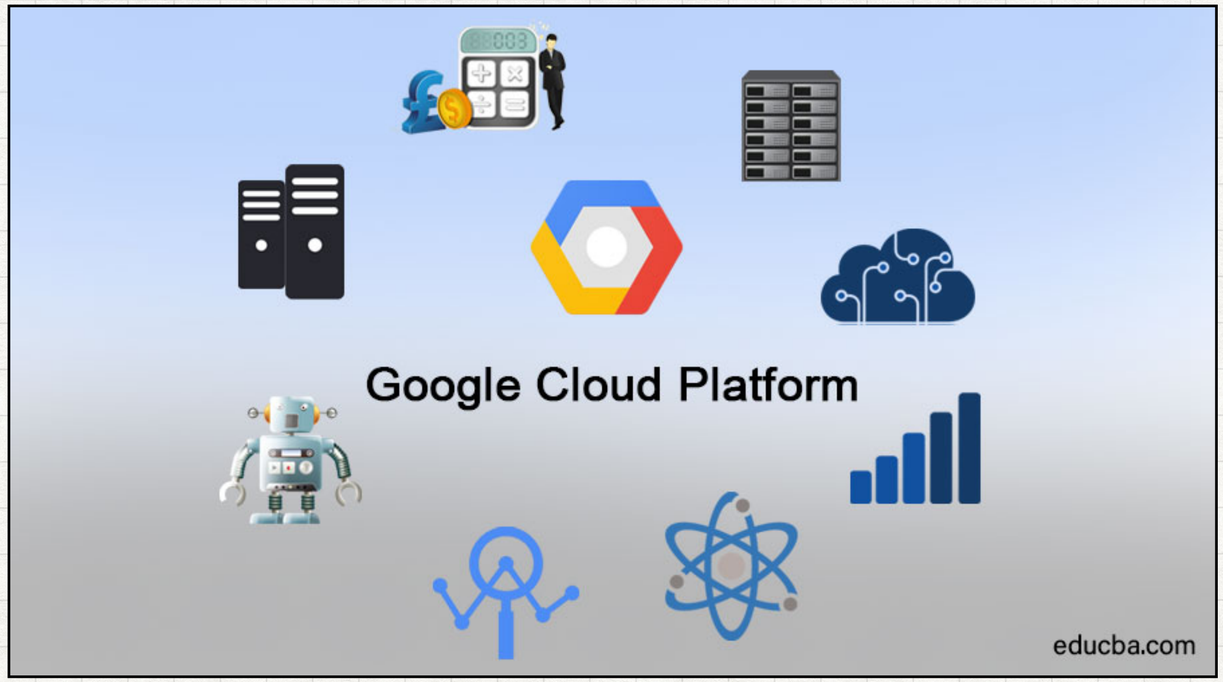
DESIGN

This section will explore the process and methods developed to calculate the value of Pi. We will discuss the architecture, tools, and techniques employed to implement the calculation, ensuring an efficient and accurate approach.



TECHNOLOGY USED

- **Google Cloud Platform (GCP) Ubuntu:** Utilized as the project environment.
- **Hadoop Framework:** Employed to implement the MapReduce model for distributed processing.
- **Java Language:** Used for programming and executing the Pi calculation.



A general look alike of PI calculation in a table format

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							

PROCESS

1. **Prepare Input File**

   A. Generate Random Points:

   - Create a Java program that generates random pairs of points (x,y) within a specified radius.

   B. Save Results to File:

   - Save the generated points to a file, which will be used as the input for a MapReduce job.

PROCESS

2. MapReduce Program:

- Write a MapReduce program in Java to process the points.
- Count the number of points that fall inside and outside the circle with the given radius.
- Save the output, which will show the count of points inside and outside the circle.

PROCESS

Step 3: Run MapReduce on GCP

- Use Input File: Utilize the points.txt file generated in Step 1.
- Execute MapReduce Job: Run the MapReduce program from Step 2 on Google Cloud Platform (GCP).
- Expected Output:
    - Inside: xxx
    - Outside: xxx

PROCESS

Step 4: Calculate Pi

- Java Program: Write a Java program to calculate the value of Pi.
- Use Output: Use the results from Step 3 to compute the Pi value.

IMPLEMENTATION

Environment Setup:

- Ensure proper configuration and connection to Google Cloud Platform (GCP) for execution

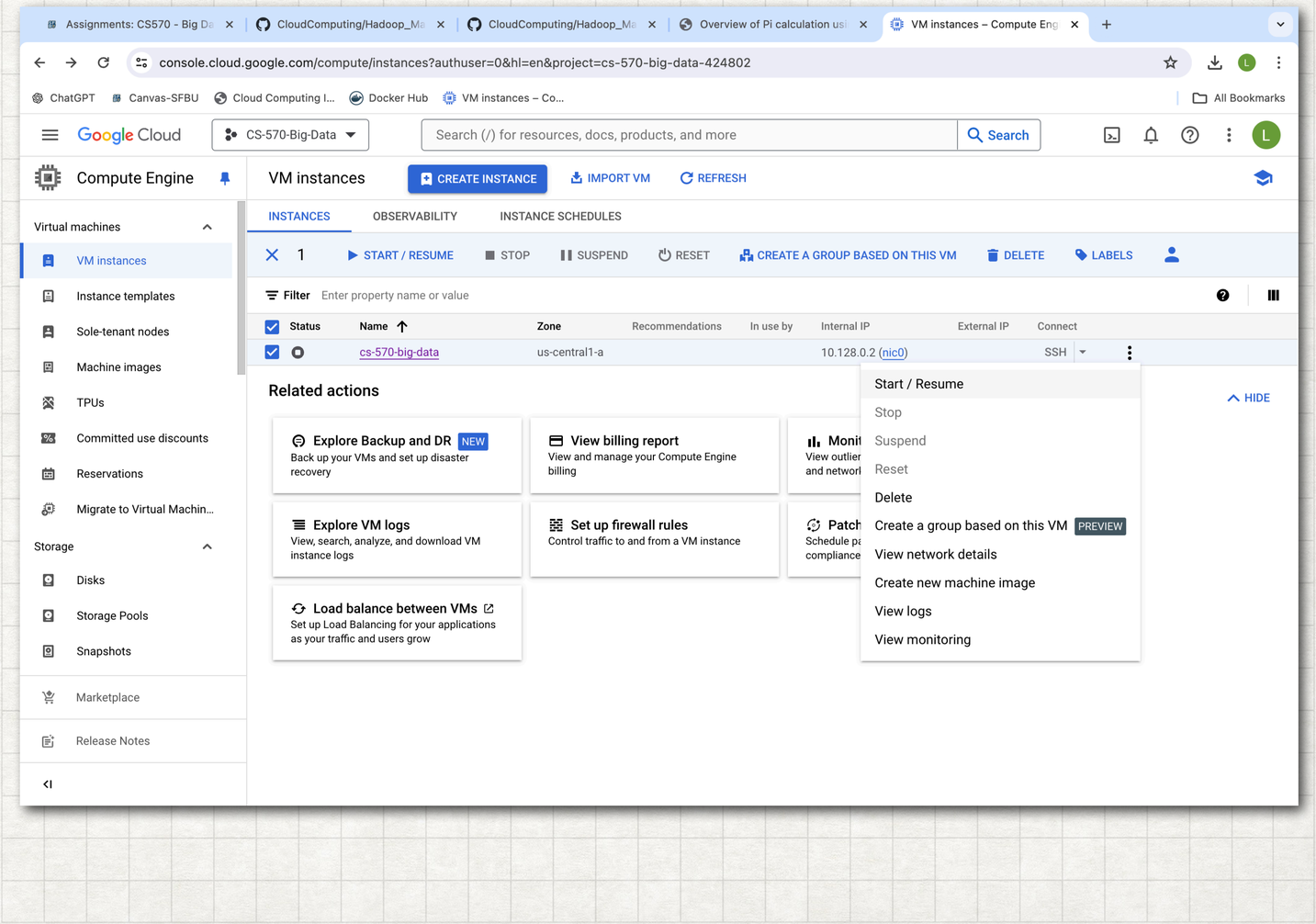
Codes

- Generate Dots.java: Create a program to generate random points.
- CalculatePiMR.java: Write a MapReduce program to count points inside and outside the circle.
- CalculatePi.java: Develop a Java program to calculate Pi using MapReduce output.

IMPLEMENTATION

ENVIRONMENT SETUP

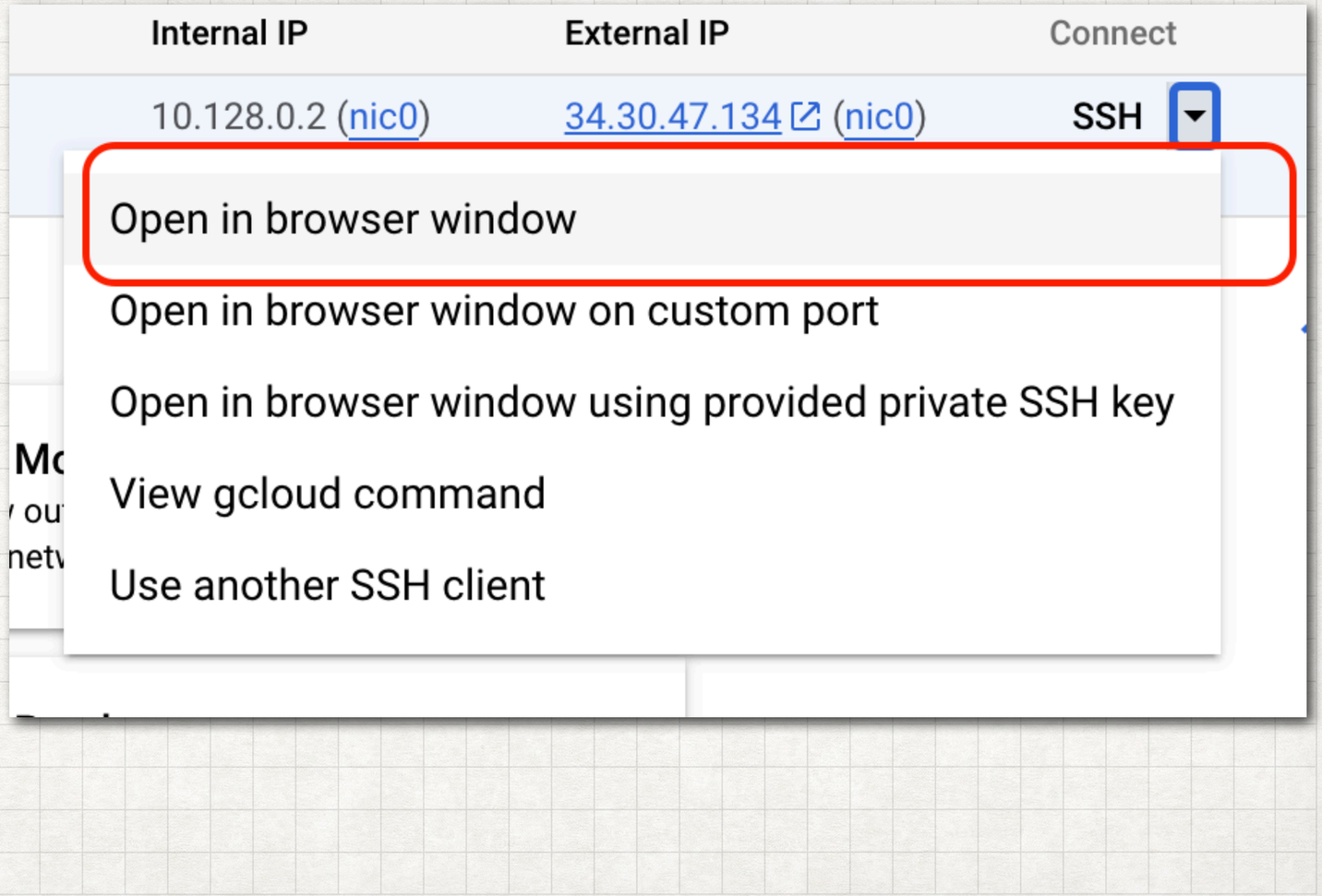
- Start VM instance in GCP



IMPLEMENTATION

ENVIRONMENT SETUP

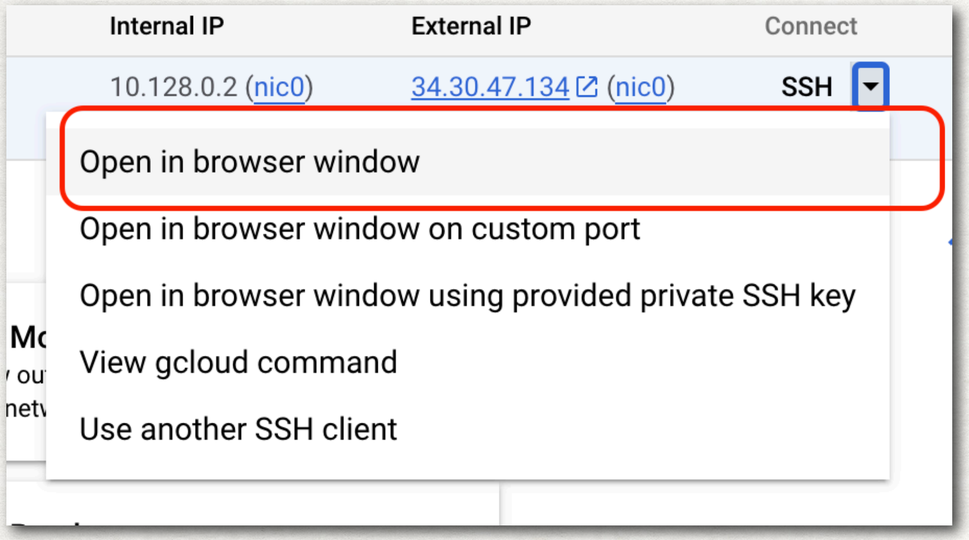
Instance Started and
Connect VM through SSH

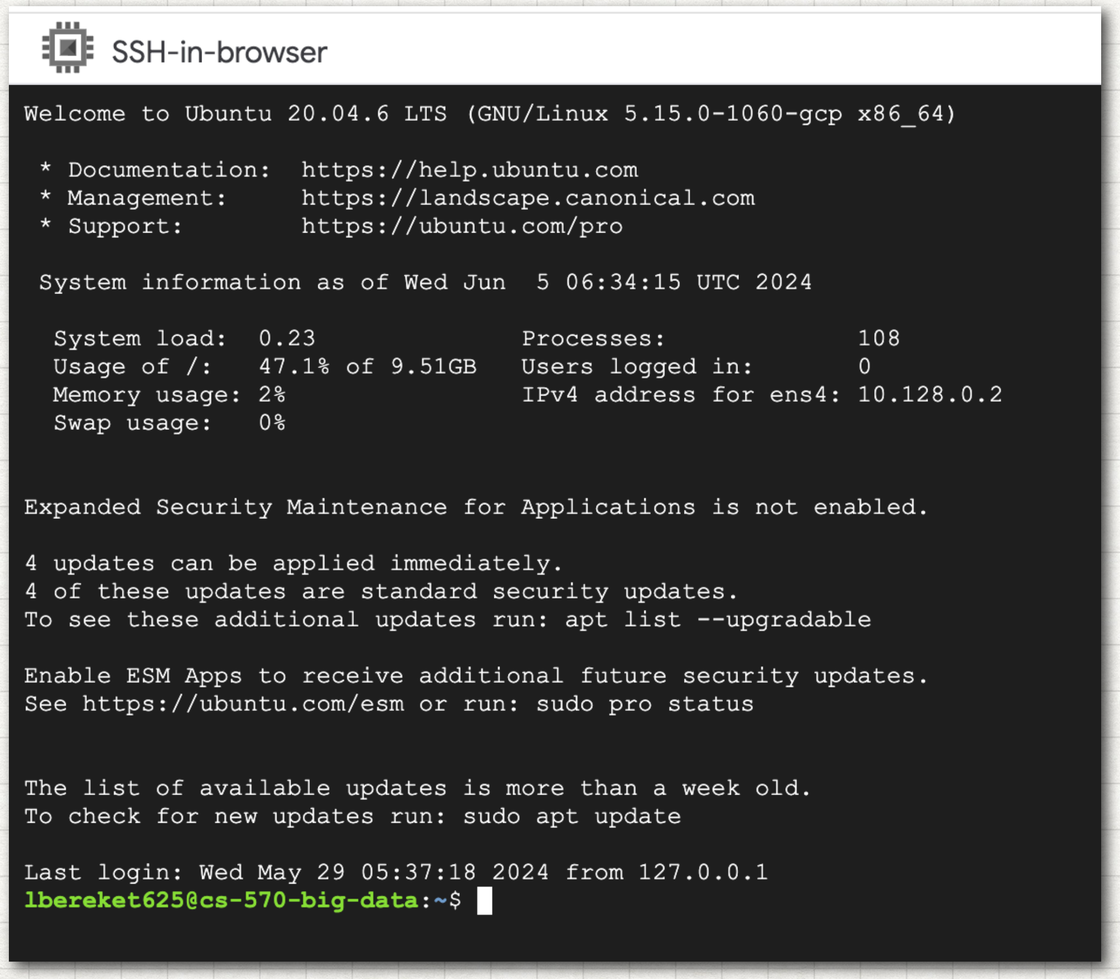


IMPLEMENTATION

ENVIRONMENT SETUP

Instance Started and
Connect VM through SSH





IMPLEMENTATION

PROGRAM/CODE SETUP

CODE—-GenerateDots.java

Java program generates random dot pairs with input radius and number of pairs, outputting coordinates in the format x y radius.

SSH-in-browser

```
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));
        }
    }
}
~
```

IMPLEMENTATION

PROGRAM/CODE SETUP

CODE—-CalculatePiMR.java

Java program this is a Mapper function for MapReduce.

SSH-in-browser

```
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="5";
                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);
            }

        }
```

IMPLEMENTATION

PROGRAM/CODE SETUP

CODE—-CalculatePiMR.java

Java program this is a Reducer function for MapReduce. And Main function for MapReduce job execution.

```
        }
    }

  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);
   }
}
```

IMPLEMENTATION

PROGRAM/CODE SETUP

CODE—-CalculatePiMR.java

Use any kind of editor, in this case vim us used to creat those programs

```
lbereket625@cs-570-big-data:~$ vim CalculatePiMR.java
lbereket625@cs-570-big-data:~$ cat CalculatePiMR.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="5";
                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);
            }
```

IMPLEMENTATION

PROGRAM/CODE SETUP

CODE—-CalculatePi.java

This Java program calculates Pi using MapReduce results obtained from file input.

SSH-in-browser

```
import java.io.*;
public 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();
        }
}
```

CODE STRUCTURE

Home Directory

- **PiProject**
  - Java Programs - all 3 of the programs
  - Input Directory - *The Input files are saved here*

- **Hadoop-3.3.5. -** *Run bin/hadoop; bin/hdfs under this directory*
  - MR.java Program - *generate *class files here*
  - Output Directory - *final output will be saved locally in this folder*

CODE STRUCTURE

- PiProject directory
- hadoop-3.3.4 directory

```
lbereket625@cs-570-big-data:~$ ls
CalculatePi.java  CalculatePiMR.java  GenerateDots.java  WordCount  hadoop-3.3.5  hadoop-3.3.5.tar.gz
lbereket625@cs-570-big-data:~$ mkdir PiProject
lbereket625@cs-570-big-data:~$ ls
CalculatePi.java  CalculatePiMR.java  GenerateDots.java  PiProject  WordCount  hadoop-3.3.5  hadoop-3.3.5.tar.gz
lbereket625@cs-570-big-data:~$ mv CalculatePi.java CalculatePiMR.java GenerateDots.java PiProject/
lbereket625@cs-570-big-data:~$ ls
PiProject  WordCount  hadoop-3.3.5  hadoop-3.3.5.tar.gz
lbereket625@cs-570-big-data:~$ cd PiProject
lbereket625@cs-570-big-data:~/PiProject$ ls
CalculatePi.java  CalculatePiMR.java  GenerateDots.java
lbereket625@cs-570-big-data:~/PiProject$
```

- Input directory

```
lbereket625@cs-570-big-data:~/PiProject$ mkdir Input
lbereket625@cs-570-big-data:~/PiProject$ ls
CalculatePi.java  CalculatePiMR.java  GenerateDots.java  Input
lbereket625@cs-570-big-data:~/PiProject$
```

TEST

PROCESS TO TEST THE PROJECT

GCP-HADOOP-MAPREDUCE

- Step 1: Execution
  - Detailed procedure of project execution and its outcomes.
- Step 2: Outcome
  - Final Pi value calculated

TEST

PROCESS TO TEST THE PROJECT

- **Format file system**

  $ bin/hdfs namenode -format
  $ sbin/start-dfs.sh

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

```
lbereket625@cs-570-big-data:~/hadoop-3.3.5$ sbin/start-dfs.sh
Starting namenodes on [localhost]
localhost: lbereket625@localhost: Permission denied (publickey).
Starting datanodes
localhost: lbereket625@localhost: Permission denied (publickey).
Starting secondary namenodes [cs-570-big-data]
cs-570-big-data: lbereket625@cs-570-big-data: Permission denied (publickey).
lbereket625@cs-570-big-data:~/hadoop-3.3.5$
```

TEST

PROCESS TO TEST THE PROJECT

$ ssh-keygen -t rsa -P '' -f ~/.ssh/id_rsa

$ cat ~/.ssh/id_rsa.pub >> ~/.ssh/

authorized_keys $ chmod 0600 ~/.ssh/

authorized_keys

$ ssh localhost

```
lbereket625@cs-570-big-data:~/hadoop-3.3.5$ ssh-keygen -t rsa -P '' -f ~/.ssh/id_rsa
Generating public/private rsa key pair.
/home/lbereket625/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Your identification has been saved in /home/lbereket625/.ssh/id_rsa
Your public key has been saved in /home/lbereket625/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:axqwskiFvUMzVFmuepEJRULQ6y7yBB5MeT/cUqwzclQ lbereket625@cs-570-big-data
The key's randomart image is:
+---[RSA 3072]----+
| .+o.+oE         |
|  ..+.+          |
| o +.. +         |
|o =.= B          |
|.+.B.# .S        |
|..+.Bo*  .       |
| o++... o        |
|o+ +o  +         |
|o.+   .          |
+----[SHA256]-----+
lbereket625@cs-570-big-data:~/hadoop-3.3.5$ cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
lbereket625@cs-570-big-data:~/hadoop-3.3.5$ chmod 0600 ~/.ssh/authorized_keys
lbereket625@cs-570-big-data:~/hadoop-3.3.5$ ssh localhost
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 07:12:05 UTC 2024

  System load:  0.03              Processes:             109
  Usage of /:   51.1% of 9.51GB   Users logged in:       1
  Memory usage: 3%                IPv4 address for ens4: 10.128.0.2
  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

New release '22.04.3 LTS' available.
Run 'do-release-upgrade' to upgrade to it.
```

TEST

PROCESS TO TEST THE PROJECT

Format again!

Successfully started!

```
lbereket625@cs-570-big-data:~/hadoop-3.3.5$ bin/hdfs namenode -format
2024-06-05 07:14:15,228 INFO namenode.NameNode: STARTUP_MSG:
/************************************************************
STARTUP_MSG: Starting NameNode
STARTUP_MSG:   host = cs-570-big-data.us-central1-a.c.cs-570-big-data-424802.internal/10.128.0.2
STARTUP_MSG:   args = [-format]
STARTUP_MSG:   version = 3.3.5
STARTUP_MSG:   classpath = /home/lbereket625/hadoop-3.3.5/etc/hadoop:/home/lbereket625/hadoop-3.3.5/share/ha
-3.3.5/share/hadoop/common/lib/netty-transport-native-kqueue-4.1.77.Final-osx-x86_64.jar:/home/lbereket625/
ome/lbereket625/hadoop-3.3.5/share/hadoop/common/lib/jetty-util-9.4.48.v20220622.jar:/home/lbereket625/hadoo
eket625/hadoop-3.3.5/share/hadoop/common/lib/jakarta.activation-api-1.2.1.jar:/home/lbereket625/hadoop-3.3.
adoop-3.3.5/share/hadoop/common/lib/netty-codec-redis-4.1.77.Final.jar:/home/lbereket625/hadoop-3.3.5/share
a/hadoop/common/lib/kerby-util-1.0.1.jar:/home/lbereket625/hadoop-3.3.5/share/hadoop/common/
common/lib/commons-text-1.10.0.jar:/home/lbereket625/hadoop-3.3.5/share/hadoop/common/lib/jaxb-api-2.2.11.j
```

Successful started!

```
lbereket625@cs-570-big-data:~/hadoop-3.3.5$ sbin/start-dfs.sh
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [cs-570-big-data]
lbereket625@cs-570-big-data:~/hadoop-3.3.5$
```

TEST

PROCESS TO TEST THE PROJECT

Test Connection with localhost

$ wget http://localhost:9870/

```
lbereket625@cs-570-big-data:~/hadoop-3.3.5$ wget http://localhost:9870/
--2024-06-05 07:16:46--  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 07:16:46--  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.4’

index.html.4                        100%[=============================================================================================>]   1.05K  --.-KB/s    in 0s

2024-06-05 07:16:46 (177 MB/s) - ‘index.html.4’ saved [1079/1079]

lbereket625@cs-570-big-data:~/hadoop-3.3.5$
```

TEST

PROCESS TO TEST THE PROJECT

$ javac GenerateDots.java

```
lbereket625@cs-570-big-data:~/hadoop-3.3.5$ cd ..
lbereket625@cs-570-big-data:~$ ls
PiProject  WordCount  hadoop-3.3.5  hadoop-3.3.5.tar.gz
lbereket625@cs-570-big-data:~$ cd PiProject
lbereket625@cs-570-big-data:~/PiProject$ ls
CalculatePi.java  CalculatePiMR.java  GenerateDots.java  Input
lbereket625@cs-570-big-data:~/PiProject$ javac GenerateDots.java
lbereket625@cs-570-big-data:~/PiProject$ ls
CalculatePi.java  CalculatePiMR.java  GenerateDots.class  GenerateDots.java  Input
lbereket625@cs-570-big-data:~/PiProject$
```

$ java GenerateDots 5 1000 > ./Input/dots.txt

```
lbereket625@cs-570-big-data:~/PiProject$ cat ./Input/dots.txt
9.61924462536099 3.3318410207338034 5.0
0.7722336398656937 8.546852791058292 5.0
1.7025353020604739 7.898836689881747 5.0
9.572739184504407 0.44814068871928736 5.0
1.8546065767474096 0.2606355217627532 5.0
9.252098302851044 1.10034599964709 5.0
8.107845536147765 6.688492899341378 5.0
6.978118120199835 7.19103308594606 5.0
6.848211850593527 9.612974435117982 5.0
2.585065588813146 2.597750519892311 5.0
8.563769899936357 6.833868852358831 5.0
7.367647779314652 3.7202144198272755 5.0
2.487233119858745 3.2783667277636 82 5.0
4.002435549065107 9.448922763529055 5.0
7.326432858318319 7.074167598753371 5.0
5.752552653416961 1.037837943289669 5.0
6.34107165871081 8.471755954274983 5.0
8.707158097428634 9.697607911364617 5.0
7.651075372377799 0.9801630465191191 5.0
2.9654472898392212 1.3371203756062844 5.0
4.851315377968439 8.973116116988852 5.0
7.395608666520248 8.55501066980435 5.0
8.759417292786601 4.977545682095258 5.0
7.882367541945192 9.719080188161067 5.0
8.437835256904066 2.8740052202372937 5.0
0.8224558896214962 3.770401369302505 5.0
8.508315093686221 0.31863687066770674 5.0
2.0355521348683414 7.250217242613934 5.0
0.21633146126940783 4.838888912142839 5.0
6.228931713239395 6.971574054362227 5.0
5.705026106533938 3.2602764000048614 5.0
3.6872805001060716 7.743948065531125 5.0
8.522642709833717 3.3890472407556427 5.0
8.954817213936959 9.60796177806764 5.0
8.514389854596944 0.2057009325931447 5.0
0.9350840454779552 3.1486076535314798 5.0
5.939130998861247 6.216877737576043 5.0
5.054457281254905 4.629228805510193 5.0
7.903244129577233 7.068224064545286 5.0
5.918642848802626 4.924833061588918 5.0
0.018453086149426978 9.038405396550317 5.0
3.2371163227172484 7.532146503714127 5.0
7.567381121139056 4.614530068111611 5.0
4.698677565726727 1.2951367628153188 5.0
```

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

TEST

PROCESS TO TEST THE PROJECT

$ bin/hdfs dfs -mkdir /user/lbereket625/PiProje
Input
$ bin/hdfs dfs -put ../PiProject/Input/* PiProject/
Input
$ bin/hdfs dfs -ls PiProject/Input

```
lbereket625@cs-570-big-data:~$ cd hadoop-3.3.5
lbereket625@cs-570-big-data:~/hadoop-3.3.5$ bin/hdfs dfs -mkdir /user
lbereket625@cs-570-big-data:~/hadoop-3.3.5$ bin/hdfs dfs -mkdir /user/lbereket625
lbereket625@cs-570-big-data:~/hadoop-3.3.5$ bin/hdfs dfs -mkdir /user/lbereket625/PiProject
lbereket625@cs-570-big-data:~/hadoop-3.3.5$ bin/hdfs dfs -mkdir /user/lbereket625/PiProject/Input
lbereket625@cs-570-big-data:~/hadoop-3.3.5$
```

Copy file from local to hadoop and check

```
lbereket625@cs-570-big-data:~/hadoop-3.3.5$ bin/hdfs dfs -put ../PiProject/Input/* PiProject/Input
lbereket625@cs-570-big-data:~/hadoop-3.3.5$ bin/hdfs dfs -ls PiProject/Input
Found 1 items
-rw-r--r--   1 lbereket625 supergroup      40536 2024-06-05 07:25 PiProject/Input/dots.txt
lbereket625@cs-570-big-data:~/hadoop-3.3.5$
```

TEST

PROCESS TO TEST THE PROJECT

$ bin/hadoop com.sun.tools.javac.Main ./ CalculatePiMR.java

```
lbereket625@cs-570-big-data:~/hadoop-3.3.5$ bin/hadoop com.sun.tools.javac.Main ./CalculatePiMR.java
Note: ./CalculatePiMR.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
lbereket625@cs-570-big-data:~/hadoop-3.3.5$
```

Compile Mapreduce program in Hadoop with *.class files created

```
lbereket625@cs-570-big-data:~/hadoop-3.3.5$ ls
'CalculatePiMR$Map.class'      LICENSE-binary   README.txt                        WordCount.java   index.html     index.html.4   licenses-binary   sbin
'CalculatePiMR$Reduce.class'   LICENSE.txt      'WordCount$IntSumReducer.class'   bin              index.html.1   input          logs              share
 CalculatePiMR.class           NOTICE-binary    'WordCount$TokenizerMapper.class' etc              index.html.2   lib            output            wc.jar
 CalculatePiMR.java            NOTICE.txt        WordCount.class                  include          index.html.3   libexec        output1
lbereket625@cs-570-big-data:~/hadoop-3.3.5$
```

TEST

PROCESS TO TEST THE PROJECT

$ jar cf pi.jar CalculatePiMR*.class

Create .jar file with *.class files

```
lbereket625@cs-570-big-data:~/hadoop-3.3.5$ jar cf pi.jar CalculatePiMR*.class
lbereket625@cs-570-big-data:~/hadoop-3.3.5$ ls
'CalculatePiMR$Map.class'      LICENSE-binary   README.txt                        WordCount.java   index.html     index.html.4   licenses-binary   pi.jar
'CalculatePiMR$Reduce.class'   LICENSE.txt      'WordCount$IntSumReducer.class'   bin              index.html.1   input          logs              sbin
 CalculatePiMR.class           NOTICE-binary   'WordCount$TokenizerMapper.class'  etc              index.html.2   lib            output            share
 CalculatePiMR.java            NOTICE.txt       WordCount.class                  include          index.html.3   libexec        output1           wc.jar
lbereket625@cs-570-big-data:~/hadoop-3.3.5$
```

TEST

PROCESS TO TEST THE PROJECT

$ bin/hadoop jar pi.jar CalculatePiMR /user/lbereket625/PiProject/Input /user/lbereket625/PiProject/Output

Run MapReduce Program with input file and save result in Output

```
lbereket625@cs-570-big-data:~/hadoop-3.3.5$ jar cf pi.jar CalculatePiMR*.class
lbereket625@cs-570-big-data:~/hadoop-3.3.5$  bin/hadoop jar pi.jar CalculatePiMR /user/lbereket625/PiProject/Input /user/lbereket625/PiProject/Output
2024-06-05 07:40:27,679 INFO impl.MetricsConfig: Loaded properties from hadoop-metrics2.properties
2024-06-05 07:40:27,792 INFO impl.MetricsSystemImpl: Scheduled Metric snapshot period at 10 second(s).
2024-06-05 07:40:27,792 INFO impl.MetricsSystemImpl: JobTracker metrics system started
2024-06-05 07:40:28,014 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRun
ner to remedy this.
2024-06-05 07:40:28,223 INFO input.FileInputFormat: Total input files to process : 1
2024-06-05 07:40:28,258 INFO mapreduce.JobSubmitter: number of splits:1
2024-06-05 07:40:28,490 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_local1687929072_0001
2024-06-05 07:40:28,490 INFO mapreduce.JobSubmitter: Executing with tokens: []
2024-06-05 07:40:28,706 INFO mapreduce.Job: The url to track the job: http://localhost:8080/
2024-06-05 07:40:28,707 INFO mapreduce.Job: Running job: job_local1687929072_0001
2024-06-05 07:40:28,724 INFO mapred.LocalJobRunner: OutputCommitter set in config null
2024-06-05 07:40:28,741 INFO output.PathOutputCommitterFactory: No output committer factory defined, defaulting to FileOutputCommitterFactory
2024-06-05 07:40:28,743 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 2
2024-06-05 07:40:28,743 INFO output.FileOutputCommitter: FileOutputCommitter skip cleanup _temporary folders under output directory:false, ignore cleanup failures: false
2024-06-05 07:40:28,745 INFO mapred.LocalJobRunner: OutputCommitter is org.apache.hadoop.mapreduce.lib.output.FileOutputCommitter
2024-06-05 07:40:28,826 INFO mapred.LocalJobRunner: Waiting for map tasks
2024-06-05 07:40:28,827 INFO mapred.LocalJobRunner: Starting task: attempt_local1687929072_0001_m_000000_0
2024-06-05 07:40:28,862 INFO output.PathOutputCommitterFactory: No output committer factory defined, defaulting to FileOutputCommitterFactory
2024-06-05 07:40:28,863 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 2
2024-06-05 07:40:28,863 INFO output.FileOutputCommitter: FileOutputCommitter skip cleanup _temporary folders under output directory:false, ignore cleanup failures: false
2024-06-05 07:40:28,883 INFO mapred.Task:  Using ResourceCalculatorProcessTree : [ ]
2024-06-05 07:40:28,890 INFO mapred.MapTask: Processing split: hdfs://localhost:9000/user/lbereket625/PiProject/Input/dots.txt:0+40536
2024-06-05 07:40:28,996 INFO mapred.MapTask: (EQUATOR) 0 kvi 26214396(104857584)
2024-06-05 07:40:28,996 INFO mapred.MapTask: mapreduce.task.io.sort.mb: 100
2024-06-05 07:40:28,996 INFO mapred.MapTask: soft limit at 83886080
2024-06-05 07:40:28,996 INFO mapred.MapTask: bufstart = 0; bufvoid = 104857600
2024-06-05 07:40:28,996 INFO mapred.MapTask: kvstart = 26214396; length = 6553600
2024-06-05 07:40:29,005 INFO mapred.MapTask: Map output collector class = org.apache.hadoop.mapred.MapTask$MapOutputBuffer
2024-06-05 07:40:29,220 INFO mapred.LocalJobRunner:
2024-06-05 07:40:29,230 INFO mapred.MapTask: Starting flush of map output
2024-06-05 07:40:29,231 INFO mapred.MapTask: Spilling map output
2024-06-05 07:40:29,231 INFO mapred.MapTask: bufstart = 0; bufend = 11191; bufvoid = 104857600
2024-06-05 07:40:29,231 INFO mapred.MapTask: kvstart = 26214396(104857584); kvend = 26210400(104841600); length = 3997/6553600
2024-06-05 07:40:29,253 INFO mapred.MapTask: Finished spill 0
2024-06-05 07:40:29,267 INFO mapred.Task: Task:attempt_local1687929072_0001_m_000000_0 is done. And is in the process of committing
2024-06-05 07:40:29,273 INFO mapred.LocalJobRunner: map
2024-06-05 07:40:29,274 INFO mapred.Task: Task 'attempt_local1687929072_0001_m_000000_0' done.
2024-06-05 07:40:29,285 INFO mapred.Task: Final Counters for attempt_local1687929072_0001_m_000000_0: Counters: 23
        File System Counters
                FILE: Number of bytes read=3509
                FILE: Number of bytes written=664215
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
```

TEST

PROCESS TO TEST THE PROJECT

$ bin/hdfs dfs -get PiProject/Output Output

Get output and save to local

```
Bytes Written=23
lbereket625@cs-570-big-data:~/hadoop-3.3.5$ bin/hdfs dfs -get PiProject/Output Output
lbereket625@cs-570-big-data:~/hadoop-3.3.5$
```

$ cat Output/*

Display Output

```
lbereket625@cs-570-big-data:~/hadoop-3.3.5$ cat Output/*
Inside  809
Outside 191
lbereket625@cs-570-big-data:~/hadoop-3.3.5$
```

RESULT

PROCESS TO TEST THE PROJECT

$ jvac CalculatePi.java
$ java CalculatePi Output

```
lbereket625@cs-570-big-data:~/PiProject$ vi  CalculatePi.java
lbereket625@cs-570-big-data:~/PiProject$ javac CalculatePi.java
lbereket625@cs-570-big-data:~/PiProject$ java CalculatePi Output
Inside  809
Outside 191
PI value is: 3.236
lbereket625@cs-570-big-data:~/PiProject$
```

Using the output (local output folder as command line arguments) from MapReduce Program to compile and run java program to get pi value
The pi value calculated is 3.236, and it is a bit off from 3.1415926 but not too much

RESULT
ENHANCEMENT

Proving Test Results

- **Base Case:**
  - Radius: 5
  - Number of dots: 1000
  - Calculated Pi value: 3.236
- **Adjusting Parameters:**
  - Decrease radius to cover more area:
    - New radius: 1
    - Number of dots: 1000
  - Increase number of dots to enhance accuracy:
    - Radius: 5
    - New number of dots: 100,000

RESULT
ENHANCEMENT--DECREASE NUMBER

radius = 1 number = 1000

```
lbereket625@cs-570-big-data:~/PiProject$ ls
CalculatePi.class  CalculatePi.java  CalculatePiMR.java  GenerateDots.class  GenerateDots.java  Input
lbereket625@cs-570-big-data:~/PiProject$ javac GenerateDots.java
lbereket625@cs-570-big-data:~/PiProject$ java GenerateDots 1 1000 > ./Input/test1.txt
lbereket625@cs-570-big-data:~/PiProject$ ls ./Input
dots.txt  test1.txt
lbereket625@cs-570-big-data:~/PiProject$ cat ./Input/test1.text
cat: ./Input/test1.text: No such file or directory
lbereket625@cs-570-big-data:~/PiProject$ cat ./Input/test1.txt
0.01992078371054773 0.12216097521049085 1.0
0.32476130057297525 1.7597350974209207 1.0
0.27421835468403644 1.229411707011077 1.0
0.4802560207732416 0.06419380488253301 1.0
1.7423875535544022 1.5024265293116064 1.0
1.0516980203791049 0.45374287287646253 1.0
1.6763363709220276 1.5192198393281204 1.0
0.15441502141963404 1.1898425412909985 1.0
1.9162889165197647 0.27862691357023395 1.0
0.04350224625812027 1.5101354996670402 1.0
1.9185561531522788 1.084531725165807 1.0
0.45657732948064633 0.9458919288596688 1.0
1.6217255911480448 0.8166252426671321 1.0
1.058779036762325 0.11817836556275085 1.0
1.1110480485610923 0.3956153745575315 1.0
1.1031073920421988 0.4193046011233108 1.0
1.5505643067997632 1.776202721949118 1.0
1.585219007008003 1.4177460297008926 1.0
1.174781295443232 0.639024860619611 1.0
1.1503074887348266 1.5688758814050368 1.0
1.2664723600244532 0.10316400940324888 1.0
1.8007132822210223 1.8704864716512533 1.0
1.994441705905626 1.2437024561449492 1.0
```

RESULT

ENHANCEMENT --DECREASE NUMBER

```
lbereket625@cs-570-big-data:~$ cd hadoop-3.3.5
lbereket625@cs-570-big-data:~/hadoop-3.3.5$ bin/hdfs dfs -put ../PiProject/Input/test1.txt PiProject/Input
lbereket625@cs-570-big-data:~/hadoop-3.3.5$ bin/hdfs dfs -ls PiProject/Input
Found 2 items
-rw-r--r--   1 lbereket625 supergroup      40536 2024-06-05 07:25 PiProject/Input/dots.txt
-rw-r--r--   1 lbereket625 supergroup      42047 2024-06-05 07:57 PiProject/Input/test1.txt
lbereket625@cs-570-big-data:~/hadoop-3.3.5$
```

```
lbereket625@cs-570-big-data:~/PiProject$ javac GenerateDots.java
lbereket625@cs-570-big-data:~/PiProject$ java GenerateDots 1 1000 > ./Input/test1.txt
lbereket625@cs-570-big-data:~/PiProject$ ls ./Input
dots.txt  test1.txt
lbereket625@cs-570-big-data:~/PiProject$ cat ./Input/test1.text
cat: ./Input/test1.text: No such file or directory
lbereket625@cs-570-big-data:~/PiProject$ cat ./Input/test1.txt
0.01992078371054773 0.12216097521049085 1.0
0.32476130057297525 1.7597350974209207 1.0
0.27421835468403644 1.229411707011077 1.0
0.4802560207732416 0.06419380488253301 1.0
1.7423875535544022 1.5024265293116064 1.0
1.0516980203791049 0.45374287287646253 1.0
1.6763363709220276 1.5192198393281204 1.0
0.15441502141963404 1.1898425412909985 1.0
1.9162889165197647 0.27862691357023395 1.0
0.04350224625812027 1.5101354996670402 1.0
1.9185561531522788 1.084531725165807 1.0
0.45657732948064633 0.9458919288596688 1.0
1.6217255911480448 0.8166252426671321 1.0
1.058779036762325 0.11817836556275085 1.0
1.1110480485610923 0.3956153745575315 1.0
1.1031073920421988 0.4193046011233108 1.0
1.5505643067997632 1.776202721949118 1.0
1.585219007008003 1.4177460297008926 1.0
1.174781295443232 0.639024860619611 1.0
1.1503074887348266 1.5688758814050368 1.0
```

RESULT
ENHANCEMENT -DECREASE NUMBER

Pi value calculate is 3.144 which is a better value to the real pi value then the base case value

```
lbereket625@cs-570-big-data:~/hadoop-3.3.5$ bin/hdfs dfs -get /user/lbereket625/PiProject/Test1 Test1
lbereket625@cs-570-big-data:~/hadoop-3.3.5$ cat Test1/*
Inside  786
Outside 214
lbereket625@cs-570-big-data:~/hadoop-3.3.5$
```

```
lbereket625@cs-570-big-data:~$ cd PiProject
lbereket625@cs-570-big-data:~/PiProject$ ls
CalculatePi.class  CalculatePi.java  CalculatePiMR.java  GenerateDots.class  GenerateDots.java  Input
lbereket625@cs-570-big-data:~/PiProject$ java CalculatePi Test1
Inside  786
Outside 214
PI value is: 3.144
lbereket625@cs-570-big-data:~/PiProject$
```

RESULT
ENHANCEMENT --INCREASE NUMBER

radius = 5 number = 100000

```
lbereket625@cs-570-big-data:~/PiProject$ java GenerateDots 5 1000000 > ./Input/test2.txt
lbereket625@cs-570-big-data:~/PiProject$ ls ./Input
dots.txt  test1.txt  test2.txt
lbereket625@cs-570-big-data:~/PiProject$
```

```
lbereket625@cs-570-big-data:~/PiProject$ cat ./Input/test2.txt
8.064249911685666 9.564788191218476 5.0
1.2464931841457771 2.1707118559588423 5.0
8.9686513993712 4.892149935148204 5.0
4.679107499681365 6.171760503689283 5.0
7.027100352001272 9.776843076828184 5.0
3.69321902985845 2.8107408214493 5.0
5.114319398675367 3.30072577062271 5.0
7.97610507606053 4.546952423580745 5.0
0.3901070924785466 9.331714917038475 5.0
3.21395001106501 1.924397238379033 5.0
0.33686676839085217 3.4130725276560794 5.0
2.9149400864221384 8.992794585354723 5.0
1.6372634028768673 8.593665618877699 5.0
7.926153754559923 2.6953545445609404 5.0
0.468973160595052 7.811808160291992 5.0
1.2495337192943867 4.3095580958211706 5.0
4.053258655363539 7.720688669265279 5.0
4.67824161235222 2.2088193292189198 5.0
2.3663629728489353 6.431269950759841 5.0
5.503214700519672 1.707062981194869 5.0
5.807488659079435 5.5489481816896475 5.0
6.080751456956291 9.554873173859402 5.0
9.314385561593713 5.0931379913089 5.0
6.40548181459795 7.775024477963735 5.0
5.927345507307683 8.414352053591514 5.0
9.99907854053032 3.9568391480850673 5.0
3.1246580403413504 3.0612237278514423 5.0
6.403594754131005 2.8729551393505712 5.0
3.3873552888614986 2.700321481249092 5.0
9.35931774964988 7.338735626242544 5.0
```

RESULT
ENHANCEMENT - INCREASE NUMBER

Pi value calculate is 3.141936 which is very close to the real pi value

```
lbereket625@cs-570-big-data:~/hadoop-3.3.5$ bin/hdfs dfs -put ../PiProject/Input/test2.txt PiProject/Input
lbereket625@cs-570-big-data:~/hadoop-3.3.5$ bin/hdfs dfs -ls PiProject/Input
Found 3 items
-rw-r--r--   1 lbereket625 supergroup      40536 2024-06-05 07:25 PiProject/Input/dots.txt
-rw-r--r--   1 lbereket625 supergroup      42047 2024-06-05 07:57 PiProject/Input/test1.txt
-rw-r--r--   1 lbereket625 supergroup   40537117 2024-06-05 08:32 PiProject/Input/test2.txt
lbereket625@cs-570-big-data:~/hadoop-3.3.5$
```

```
lbereket625@cs-570-big-data:~/hadoop-3.3.5$ bin/hdfs dfs -get /user/lbereket625/PiProject/Test2 Test2
lbereket625@cs-570-big-data:~/hadoop-3.3.5$ cat Test2/*
Inside  785484
Outside 214516
lbereket625@cs-570-big-data:~/hadoop-3.3.5$
```

```
lbereket625@cs-570-big-data:~$ cd PiProject
lbereket625@cs-570-big-data:~/PiProject$ ls
CalculatePi.class  CalculatePi.java  CalculatePiMR.java  GenerateDots.class  GenerateDots.java  Input
lbereket625@cs-570-big-data:~/PiProject$ java CalculatePi Test2
Inside  785484
Outside 214516
PI value is: 3.141936
lbereket625@cs-570-big-data:~/PiProject$
```

FINAL
STOP INSTANCE ON GCP

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

```
lbereket625@cs-570-big-data:~/hadoop-3.3.5$ sbin/stop-dfs.sh
Stopping namenodes on [localhost]
Stopping datanodes
Stopping secondary namenodes [cs-570-big-data]
lbereket625@cs-570-big-data:~/hadoop-3.3.5$
```

CONCLUSION

- **Calculating Pi:**
  - Utilizing MapReduce methodology with Hadoop on Google Cloud Platform (GCP).
  - Adjusting parameters like radius and number of dots to refine accuracy.
- **Increasing Number of Dots:**
  - Enhances precision and convergence towards the true Pi value.
- **Decreasing Radius:**
  - Covers more area, providing broader data coverage for analysis.

These approaches facilitate a robust evaluation of Pi calculation results.

REFERENCE

1. Chang, H. (2022, October 09). Overview of Pi Calculation. Overview of Pi Calculation.

2. Ling, Z. (n.d.). Cloud-Computing/MapReduce/Pi. GitHub.

3. Chen, L. (2022). CS570_week2_q1_19632_Ling_Chen-1.pdf. SFBu.

My Github Link:

https://github.com/LidiaYon/Cloud-Computing-Infrastructure-/tree/main/Hadoop_MapReduce/PiProject