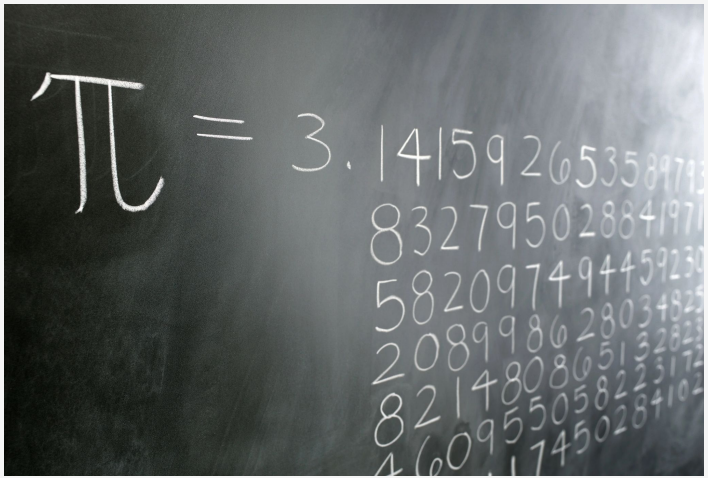
# Project: Creating MapReduce program to calculating Pi




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

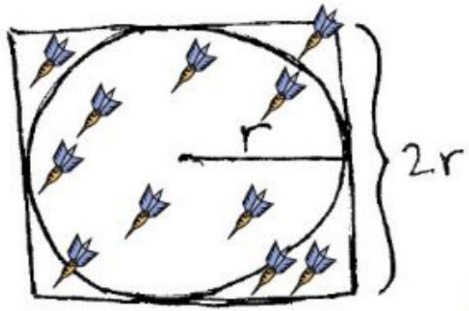
# Table of contents

# 01 Introduction


This Pi Project is to use Google Cloud Platform to
**Step 1:** Implement Hadoop with MapReduce to calculate the Pi value.
**Step 2:** Enhance the Pi calculation by transitioning to PySpark for improved performance and accuracy.

# 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 $S$




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

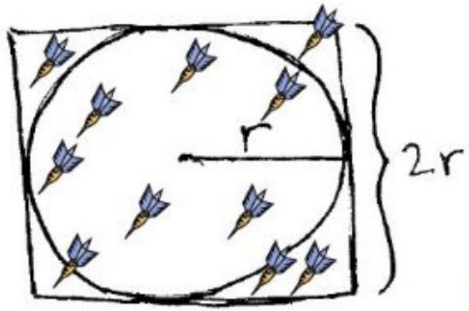
**Formula:**

**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

# 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

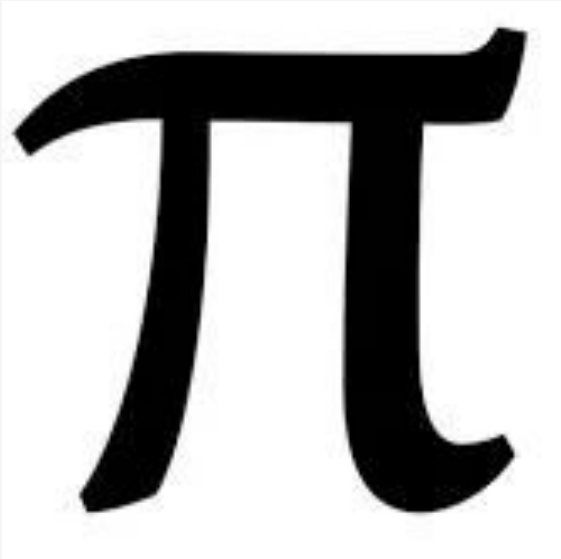
- Using GCP Ubuntu as project environment.
- Using Hadoop framework to implement MapReduce model.
- Program in Java Language.

| 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 | | | | | | | |

# 03

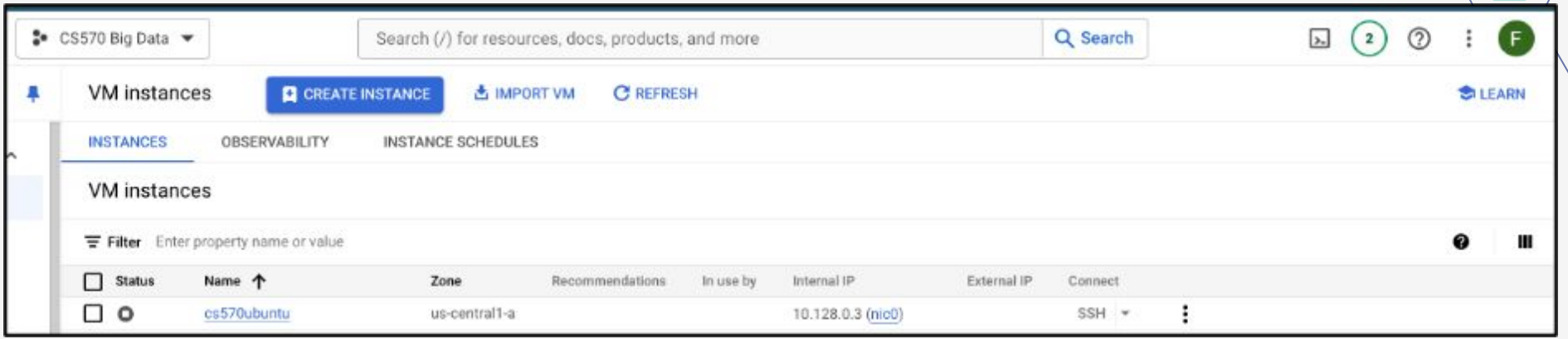
# Implementation

Getting ready to test



# Step 1: Project: Creating MapReduce program to calculating Pi

# ENVIRONMENT—-GCP



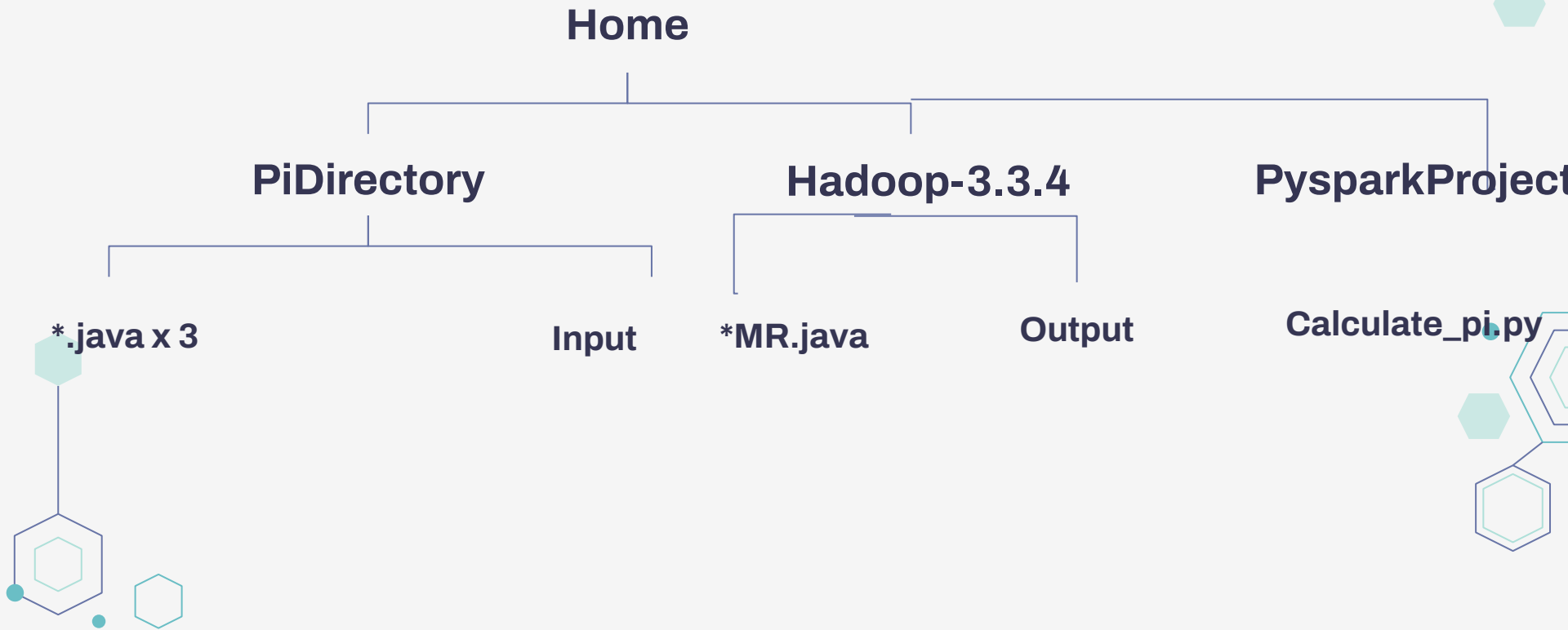

Start VM instance on GCP

# 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
```

# Step 2. Pi Calculation using PySpark

# Create a New Directory for PySpark

```
faraya85431@cs570ubuntu:~$ mkdir PySparkPiProject
faraya85431@cs570ubuntu:~$ cd PySparkPiProject
```

You can create a new directory specifically for your PySpark Pi calculation project. Let's call it PySparkPiProject for clarity:

# Install PySpark, Download and Extract Spark

```
faraya85431@cs570ubuntu:~/PySparkPiProject$ wget https://archive.apache.org/dist/spark/spark-3.1.2/spark-3.1.2-bin-hadoop3.2.tgz
--2024-06-19 03:36:39--  https://archive.apache.org/dist/spark/spark-3.1.2/spark-3.1.2-bin-hadoop3.2.tgz
Resolving archive.apache.org (archive.apache.org)... 65.108.204.189, 2a01:4f9:1a:a084::2
Connecting to archive.apache.org (archive.apache.org)|65.108.204.189|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 228834641 (218M) [application/x-gzip]
Saving to: 'spark-3.1.2-bin-hadoop3.2.tgz'

spark-3.1.2-bin-hadoop3.2.tgz               100%[=========================================================================================>] 218.23M  14.8MB/s    in 14s

2024-06-19 03:36:54 (15.3 MB/s) - 'spark-3.1.2-bin-hadoop3.2.tgz' saved [228834641/228834641]

faraya85431@cs570ubuntu:~/PySparkPiProject$ tar xvf spark-3.1.2-bin-hadoop3.2.tgz
spark-3.1.2-bin-hadoop3.2/
spark-3.1.2-bin-hadoop3.2/R/
spark-3.1.2-bin-hadoop3.2/R/lib/
spark-3.1.2-bin-hadoop3.2/R/lib/sparkr.zip
spark-3.1.2-bin-hadoop3.2/R/lib/SparkR/
spark-3.1.2-bin-hadoop3.2/R/lib/SparkR/worker/
spark-3.1.2-bin-hadoop3.2/R/lib/SparkR/worker/worker.R
spark-3.1.2-bin-hadoop3.2/R/lib/SparkR/worker/daemon.R
spark-3.1.2-bin-hadoop3.2/R/lib/SparkR/tests/
spark-3.1.2-bin-hadoop3.2/R/lib/SparkR/tests/testthat/
spark-3.1.2-bin-hadoop3.2/R/lib/SparkR/tests/testthat/test_basic.R
spark-3.1.2-bin-hadoop3.2/R/lib/SparkR/profile/
spark-3.1.2-bin-hadoop3.2/R/lib/SparkR/profile/shell.R
spark-3.1.2-bin-hadoop3.2/R/lib/SparkR/profile/general.R
spark-3.1.2-bin-hadoop3.2/R/lib/SparkR/doc/
spark-3.1.2-bin-hadoop3.2/R/lib/SparkR/doc/sparkr-vignettes.html
spark-3.1.2-bin-hadoop3.2/R/lib/SparkR/doc/sparkr-vignettes.Rmd
spark-3.1.2-bin-hadoop3.2/R/lib/SparkR/doc/sparkr-vignettes.R
spark-3.1.2-bin-hadoop3.2/R/lib/SparkR/doc/index.html
spark-3.1.2-bin-hadoop3.2/R/lib/SparkR/R/
spark-3.1.2-bin-hadoop3.2/R/lib/SparkR/R/SparkR
spark-3.1.2-bin-hadoop3.2/R/lib/SparkR/R/SparkR.rdx
spark-3.1.2-bin-hadoop3.2/R/lib/SparkR/R/SparkR.rdb
spark-3.1.2-bin-hadoop3.2/R/lib/SparkR/Meta/
spark-3.1.2-bin-hadoop3.2/R/lib/SparkR/Meta/features.rds
spark-3.1.2-bin-hadoop3.2/R/lib/SparkR/Meta/package.rds
```

```
faraya85431@cs570ubuntu:~/PySparkPiProject$ sudo mv spark-3.1.2-bin-hadoop3.2 /opt/spark
faraya85431@cs570ubuntu:~/PySparkPiProject$
```

# Set Environment Variables

```
faraya85431@cs570ubuntu:~/PySparkPiProject$ echo "export SPARK_HOME=/opt/spark" >> ~/.bashrc
faraya85431@cs570ubuntu:~/PySparkPiProject$ echo "export PATH=$PATH:/opt/spark/bin:/opt/spark/sbin" >> ~/.bashrc
faraya85431@cs570ubuntu:~/PySparkPiProject$ echo "export PYSPARK_PYTHON=python3" >> ~/.bashrc
faraya85431@cs570ubuntu:~/PySparkPiProject$ source ~/.bashrc
```

You can create a new directory specifically for your PySpark Pi calculation project. Let's call it PySparkPiProject for clarity:

# Transfer or Create Your PySpark Script

```python
import argparse
import logging
from operator import add
from random import random

from pyspark.sql import SparkSession

logger = logging.getLogger(__name__)
logging.basicConfig(level=logging.INFO, format='%(levelname)s: %(message)s')

def calculate_pi(partitions, output_uri):
    """
    Calculates pi by testing a large number of random numbers against a unit circle
    inscribed inside a square. The trials are partitioned so they can be run in
    parallel on cluster instances.

    :param partitions: The number of partitions to use for the calculation.
    :param output_uri: The URI where the output is written, typically an Amazon S3
                       bucket, such as 's3://example-bucket/pi-calc'.
    """

    def calculate_hit(_):
        x = random() * 2 - 1
        y = random() * 2 - 1
        return 1 if x**2 + y**2 < 1 else 0

    tries = 100000 * partitions

    logger.info("Calculating pi with a total of %s tries in %s partitions.", tries, partitions)

    with SparkSession.builder.appName("My PyPi").getOrCreate() as spark:
        hits = spark.sparkContext.parallelize(range(tries), partitions)\
            .map(calculate_hit)\
            .reduce(add)
        pi = 4.0 * hits / tries

        logger.info("%s tries and %s hits gives pi estimate of %s.", tries, hits, pi)

        if output_uri is not None:
            df = spark.createDataFrame([(tries, hits, pi)], ['tries', 'hits', 'pi'])

if __name__ == "__main__":
    parser = argparse.ArgumentParser()
    parser.add_argument(
        '--partitions', default=2, type=int,
        help="The number of parallel partitions to use when calculating pi.")
    parser.add_argument(
        '--output_uri', help="The URI where output is saved, typically an S3 bucket.")
    args = parser.parse_args()

    calculate_pi(args.partitions, args.output_uri)
```

- Place your PySpark script into this PySparkPiProject directory. You can use nano or vim to create or edit your Python script:

# 04

# Test

Process to test the project

# Step 1: Project: Creating MapReduce program to calculating Pi

# 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*

# Step 2. Pi Calculation using PySpark

# Run Your PySpark Script

```
faraya85431@cs570ubuntu:~/PySparkPiProject/spark-3.1.2-bin-hadoop3.2$ spark-submit --master local[4] calculate_pi.py --partitions 4 --output_uri 'gs://dataproc-temp-us-east1-794524
655341-0y1diypu/output'
24/06/19 04:13:44 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
INFO: Calculating pi with a total of 400000 tries in 4 partitions.
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
24/06/19 04:13:45 INFO SparkContext: Running Spark version 3.1.2
24/06/19 04:13:45 INFO ResourceUtils: ==============================================================
24/06/19 04:13:45 INFO ResourceUtils: No custom resources configured for spark.driver.
24/06/19 04:13:45 INFO ResourceUtils: ==============================================================
24/06/19 04:13:45 INFO SparkContext: Submitted application: My PyPi
24/06/19 04:13:45 INFO ResourceProfile: Default ResourceProfile created, executor resources: Map(cores -> name: cores, amount: 1, script: , vendor: , memory -> name: memory, amount
: 1024, script: , vendor: , offHeap -> name: offHeap, amount: 0, script: , vendor: ), task resources: Map(cpus -> name: cpus, amount: 1.0)
24/06/19 04:13:45 INFO ResourceProfile: Limiting resource is cpu
24/06/19 04:13:45 INFO ResourceProfileManager: Added ResourceProfile id: 0
24/06/19 04:13:45 INFO SecurityManager: Changing view acls to: faraya85431
24/06/19 04:13:45 INFO SecurityManager: Changing modify acls to: faraya85431
24/06/19 04:13:45 INFO SecurityManager: Changing view acls groups to:
24/06/19 04:13:45 INFO SecurityManager: Changing modify acls groups to:
24/06/19 04:13:45 INFO SecurityManager: SecurityManager: authentication disabled; ui acls disabled; users  with view permissions: Set(faraya85431); groups with view permissions: Se
```

```
24/06/19 04:13:52 INFO TaskSetManager: Finished task 3.0 in stage 0.0 (TID 3) in 1943 ms on cs570ubuntu.us-central1-a.c.cs570-big-data-424622.internal (executor driver) (3/4)
24/06/19 04:13:52 INFO TaskSetManager: Finished task 2.0 in stage 0.0 (TID 2) in 1947 ms on cs570ubuntu.us-central1-a.c.cs570-big-data-424622.internal (executor driver) (4/4)
24/06/19 04:13:52 INFO TaskSchedulerImpl: Removed TaskSet 0.0, whose tasks have all completed, from pool
24/06/19 04:13:52 INFO PythonAccumulatorV2: Connected to AccumulatorServer at host: 127.0.0.1 port: 32923
24/06/19 04:13:52 INFO DAGScheduler: ResultStage 0 (reduce at /home/faraya85431/PySparkPiProject/spark-3.1.2-bin-hadoop3.2/calculate_pi.py:33) finished in 2.317 s
24/06/19 04:13:52 INFO DAGScheduler: Job 0 is finished. Cancelling potential speculative or zombie tasks for this job
24/06/19 04:13:52 INFO TaskSchedulerImpl: Killing all running tasks in stage 0: Stage finished
24/06/19 04:13:52 INFO DAGScheduler: Job 0 finished: reduce at /home/faraya85431/PySparkPiProject/spark-3.1.2-bin-hadoop3.2/calculate_pi.py:33, took 2.590085 s
INFO: 400000 tries and 313812 hits gives pi estimate of 3.13812.
```

Within the PySparkPiProject directory, you can execute your script using spark-submit:

*spark-submit --master local[4] calculate_pi.py --partitions 4 --output_uri 'gs://dataproc-temp-us-east1-794524655341-0y1diypu/output'*

# Run Your PySpark Script

**spark-submit:**

- This is the command used to submit a job to Apache Spark, which can execute a Spark application.

**--master local[4]:**

This specifies the master URL for the cluster. Here, local[4] means that the job will run locally using 4 threads. This is often used for testing Spark applications on your local machine.

**calculate_pi.py:**

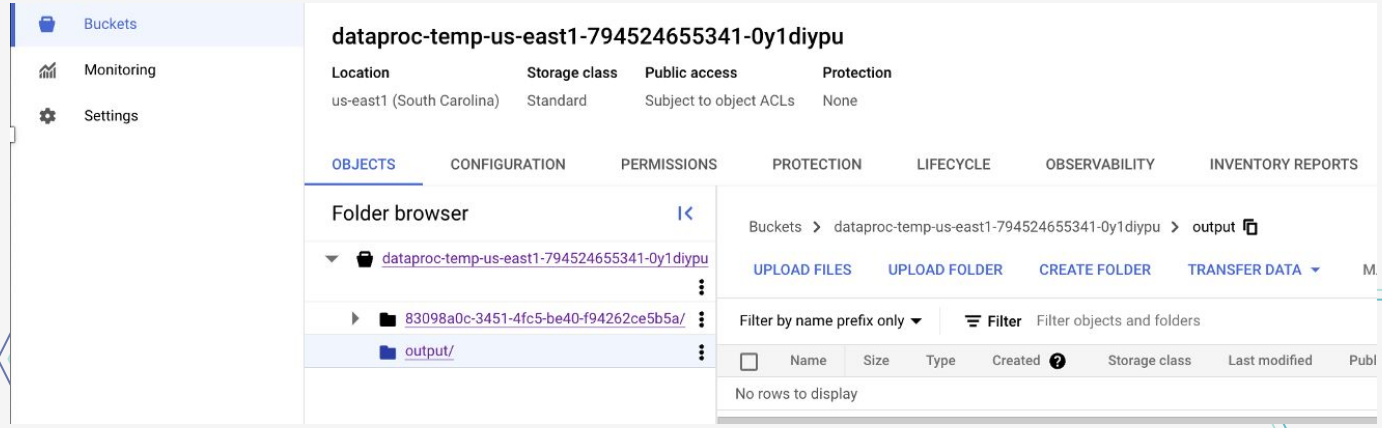
This is the Python script that is being submitted to Spark for execution. The script likely contains the Spark code to calculate the value of Pi.

**--partitions 4:**

This option is passed to the calculate_pi.py script, indicating that the dataset should be divided into 4 partitions. This can affect how the data is distributed and processed in parallel.

**--output_uri 'gs://dataproc-temp-us-east1-794524655341-0y1diypu/output':**

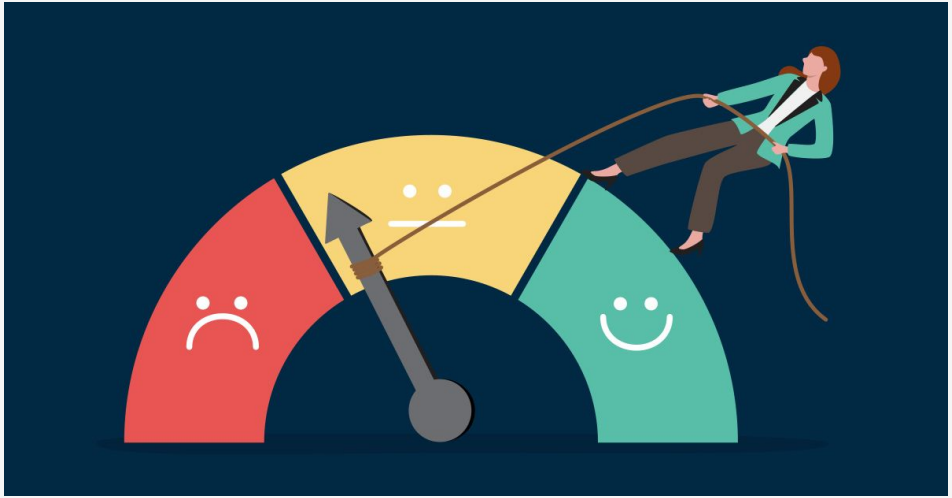
This option specifies the output location for whatever results the script produces. The URI starts with gs://, indicating that the output is to be stored on Google Cloud Storage. The rest of the URI identifies the specific bucket and path in the Google Cloud Storage where the output will be saved.



# Result

**INFO: 400000 tries and 313812 hits gives pi estimate of 3.131812.**

- This is the core result of your Pi calculation using the Monte Carlo method.
- 400000 tries refers to the number of random points generated and tested whether they fall inside a unit circle.
- 313812 hits refers to the number of points that fell inside the unit circle.
- The calculated estimate of Pi from these values is approximately 3.131812.



# 05 Enhancements

Can we get better result?

# Step 1: Project: Creating MapReduce program to calculating Pi

# 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
```

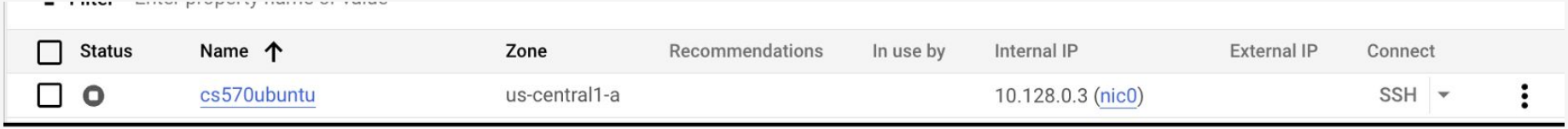
# RESULT

```
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.

# Step 2. Pi Calculation using PySpark

# Enhanced Result

```
GNU nano 4.8
from pyspark.sql import SparkSession
import random

def inside_circle(p):
    x, y = random.random(), random.random()
    return x*x + y*y < 1

if __name__ == "__main__":
    # Create Spark session
    spark = SparkSession.builder \
        .appName("Pi Calculation") \
        .config("spark.executor.memory", "2g") \
        .config("spark.driver.memory", "2g") \
        .config("spark.executor.cores", "4") \
        .getOrCreate()

    sc = spark.sparkContext

    # Number of partitions and trials
    num_partitions = 16
    num_trials = 1000000  # Increase the number of trials for better accuracy

    # Parallelize the trials into partitions
    trials = sc.parallelize(range(num_trials), num_partitions)

    # Calculate the number of points inside the circle
    count = trials.filter(inside_circle).count()

    # Estimate Pi
    pi_estimate = 4.0 * count / num_trials
    print(f"Pi is roughly estimated as {pi_estimate}")

    # Stop the Spark session
    spark.stop()
```

let's enhance the current implementation by increasing the number of partitions and the number of trials to improve the accuracy of the Pi calculation. We'll also make sure to set appropriate Spark configurations for better performance.

# Explanation

1. **Spark Configurations**:
   - Increased executor and driver memory to 2g for better resource allocation.
   - Set spark.executor.cores to 4 to utilize multiple cores for parallel processing.
2. **Increased Number of Partitions and Trials**:
   - Set num_partitions to 16 to divide the workload into more parallel tasks.
   - Increased num_trials to 1,000,000 to improve the accuracy of the Pi estimation.

# Running the Enhanced Script:

```
faraya85431@cs570ubuntu:~/PySparkPiProject$ spark-submit --master local[4] Calculate.py
24/06/19 06:17:53 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform..
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
24/06/19 06:17:54 INFO SparkContext: Running Spark version 3.1.2
24/06/19 06:17:54 INFO ResourceUtils: ==============================================================
24/06/19 06:17:54 INFO ResourceUtils: No custom resources configured for spark.driver.
24/06/19 06:17:54 INFO ResourceUtils: ==============================================================
24/06/19 06:17:54 INFO SparkContext: Submitted application: Pi Calculation
24/06/19 06:17:54 INFO ResourceProfile: Default ResourceProfile created, executor resources: Map(
: 2048, script: , vendor: , offHeap -> name: offHeap, amount: 0, script: , vendor: ), task resour
24/06/19 06:17:54 INFO ResourceProfile: Limiting resource is cpus at 4 tasks per executor
24/06/19 06:17:54 INFO ResourceProfileManager: Added ResourceProfile id: 0
24/06/19 06:17:54 INFO SecurityManager: Changing view acls to: faraya85431
24/06/19 06:17:54 INFO SecurityManager: Changing modify acls to: faraya85431
24/06/19 06:17:54 INFO SecurityManager: Changing view acls groups to:
24/06/19 06:17:54 INFO SecurityManager: Changing modify acls groups to:
24/06/19 06:17:54 INFO SecurityManager: SecurityManager: authentication disabled; ui acls disable
t(); users  with modify permissions: Set(faraya85431); groups with modify permissions: Set()
24/06/19 06:17:55 INFO Utils: Successfully started service 'sparkDriver' on port 36129.
24/06/19 06:17:55 INFO SparkEnv: Registering MapOutputTracker
24/06/19 06:17:55 INFO SparkEnv: Registering BlockManagerMaster
24/06/19 06:17:55 INFO BlockManagerMasterEndpoint: Using org.apache.spark.storage.DefaultTopology
24/06/19 06:17:55 INFO BlockManagerMasterEndpoint: BlockManagerMasterEndpoint up
```

Save the above script as calculate_pi.py and use the above command to run it:

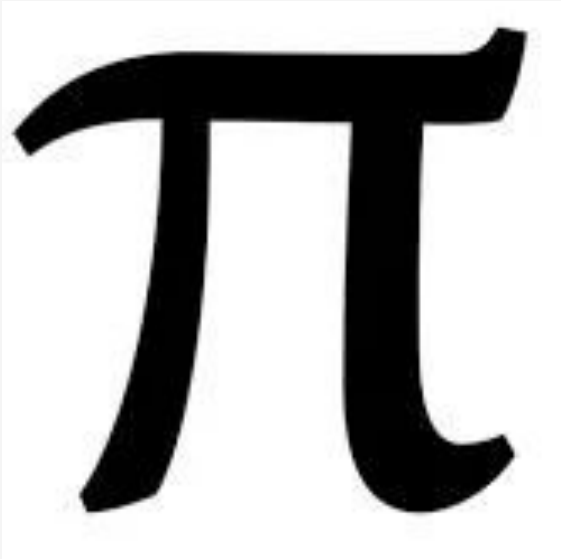
# Enhanced Result

```
24/06/19 06:18:01 INFO PythonRunner: Times: total = 230, boot = 5, init = 49, finish = 176
24/06/19 06:18:01 INFO Executor: Finished task 14.0 in stage 0.0 (TID 14). 1312 bytes result sent to driver
24/06/19 06:18:01 INFO TaskSetManager: Finished task 14.0 in stage 0.0 (TID 14) in 246 ms on cs570ubuntu.us-central1-a
24/06/19 06:18:01 INFO TaskSchedulerImpl: Removed TaskSet 0.0, whose tasks have all completed, from pool
24/06/19 06:18:01 INFO DAGScheduler: ResultStage 0 (count at /home/faraya85431/PySparkPiProject/Calculate.py:27) finis
24/06/19 06:18:01 INFO DAGScheduler: Job 0 is finished. Cancelling potential speculative or zombie tasks for this job
24/06/19 06:18:01 INFO TaskSchedulerImpl: Killing all running tasks in stage 0: Stage finished
24/06/19 06:18:01 INFO DAGScheduler: Job 0 finished: count at /home/faraya85431/PySparkPiProject/Calculate.py:27, took
Pi is roughly estimated as 3.14288
24/06/19 06:18:01 INFO SparkUI: Stopped Spark web UI at http://cs570ubuntu.us-central1-a.c.cs570-big-data-424622.inter
24/06/19 06:18:01 INFO MapOutputTrackerMasterEndpoint: MapOutputTrackerMasterEndpoint stopped!
24/06/19 06:18:01 INFO MemoryStore: MemoryStore cleared
24/06/19 06:18:01 INFO BlockManager: BlockManager stopped
24/06/19 06:18:01 INFO BlockManagerMaster: BlockManagerMaster stopped
24/06/19 06:18:01 INFO OutputCommitCoordinator$OutputCommitCoordinatorEndpoint: OutputCommitCoordinator stopped!
24/06/19 06:18:01 INFO SparkContext: Successfully stopped SparkContext
24/06/19 06:18:02 INFO ShutdownHookManager: Shutdown hook called
24/06/19 06:18:02 INFO ShutdownHookManager: Deleting directory /tmp/spark-c1eca212-ebfc-4adc-a3e2-c1354ea7ac35
24/06/19 06:18:02 INFO ShutdownHookManager: Deleting directory /tmp/spark-458e4cd9-728e-4b2c-ad83-54036c0304a1/pyspark
24/06/19 06:18:02 INFO ShutdownHookManager: Deleting directory /tmp/spark-458e4cd9-728e-4b2c-ad83-54036c0304a1
```

The output will show an improved estimation of Pi due to the increased number of trials and optimized resource usage. It should look something like:

# 06 Conclusion

Summarize for Pi Project



## Increased Random Sampling:

- More random dots enhance Pi accuracy, influenced by the circle's radius and the total number of dots.
- Step 2 increased the number of trials to improve precision.

## Efficiency of MapReduce:

- Excels at processing large datasets quickly and efficiently with minimal memory.
- Demonstrated capabilities in the initial Hadoop implementation.

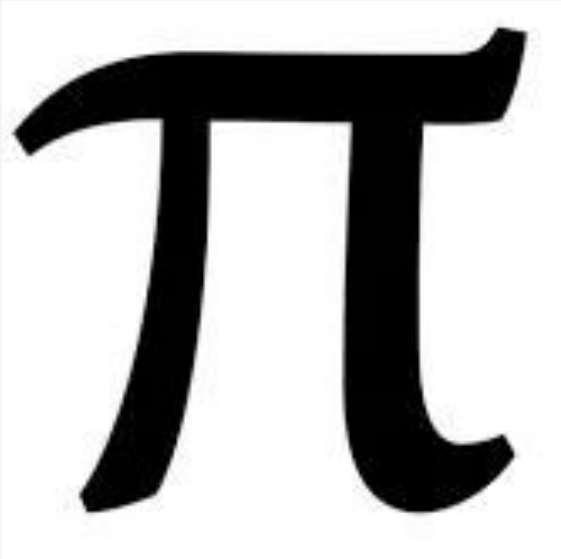
## Enhanced Calculation with PySpark:

- Transition to PySpark in Step 2 optimized Pi calculation with a flexible, efficient framework for distributed data processing.
- Increased partitions and optimized resource usage led to faster execution and improved Pi estimate accuracy.
- Utilized Spark configurations like spark.executor.memory, spark.driver.memory, and spark.executor.cores for better resource utilization.

## Overall Performance Improvement:

- Combining MapReduce and PySpark demonstrates the effective use of big data processing techniques.
- Enhancements in Step 2 highlight the importance of optimizing both the algorithm and the execution environment.

# 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.

Github Link

# Thanks!