Pi Calculation Using MapReduce & PySpark

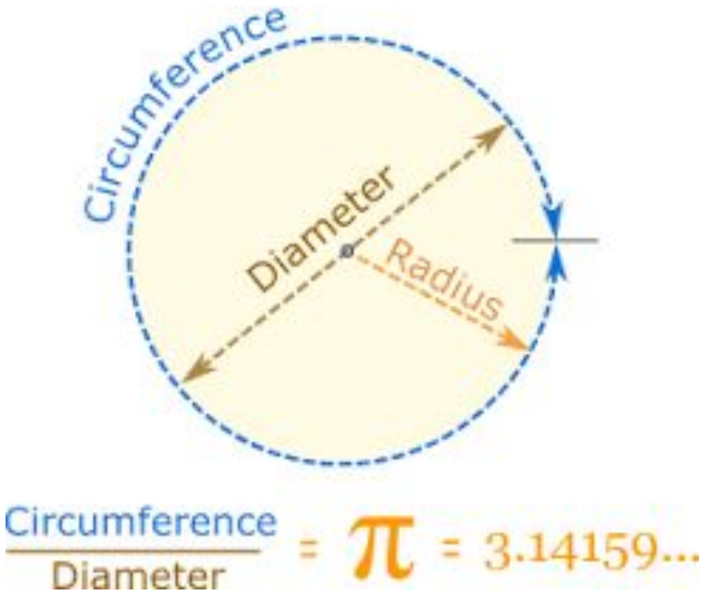
Natnael Haile
ID: 20007
June 19, 2024

Table of Contents

- Introduction
- Design
- Mapreduce
  - Implementation
  - Test
- PySpark
  - Implementation
  - Test
- Enhancement Ideas
- Conclusion
- References

Introduction

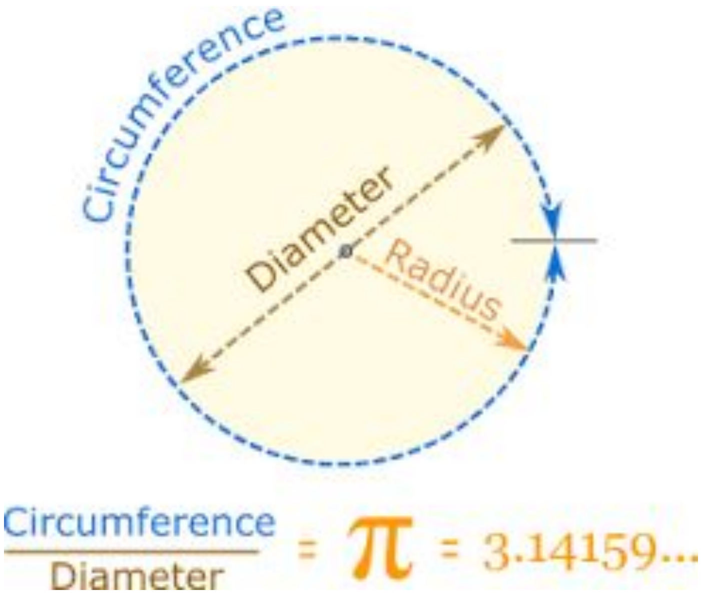
- Our primary objective in this project is to calculate the value of Pi using two distinct distributed computing frameworks:
    - Hadoop MapReduce and
    - PySpark.
- While Hadoop is typically not the first choice for computationally intensive tasks, this exercise serves as a practical application to understand and utilize Hadoop's capabilities.
- Additionally, we will leverage PySpark to illustrate its efficiency and versatility in handling similar tasks.
- By undertaking this project, we will explore and compare the frameworks' abilities to handle large-scale data processing and computational tasks.



Introduction cont.

Why Calculate Pi?

- Calculating Pi is a well-known problem in computer science and mathematics, offering a straightforward way to demonstrate the power of distributed computing.
- The simplicity of the Monte Carlo method for estimating Pi makes it an ideal candidate for this exercise.
- It involves random sampling, which can be easily parallelized, providing a clear illustration of the frameworks' data processing and computational capabilities.



Introduction cont.

Hadoop MapReduce Program

- Purpose: Demonstrate Hadoop's framework in a computational context.
- Methodology: Use the Monte Carlo method to estimate the value of Pi through a series of parallel computations distributed across a Hadoop cluster.
- Outcome: Gain hands-on experience with Hadoop's MapReduce model and its potential for handling large-scale data processing tasks.

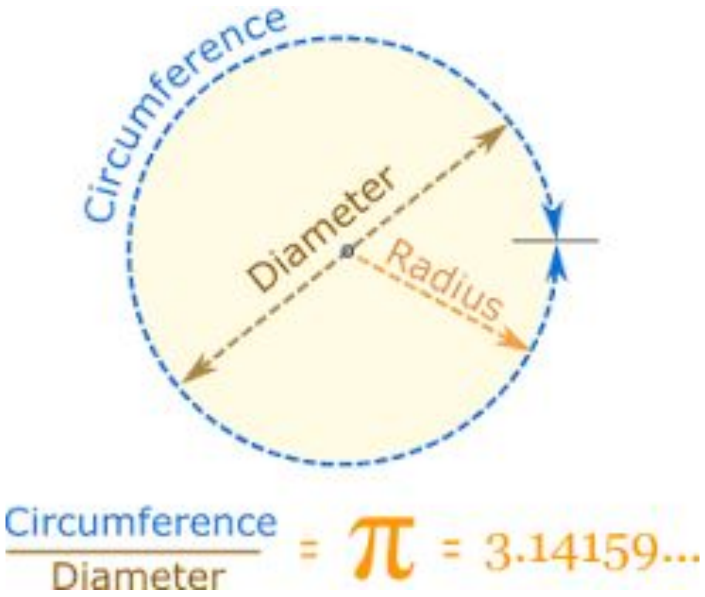
PySpark Program

- Purpose: Highlight PySpark's efficiency and ease of use in distributed computing.
- Methodology: Implement the Monte Carlo method to estimate Pi, leveraging PySpark's in-memory processing capabilities and simplified API.
- Outcome: Understand PySpark's advantages in terms of speed, scalability, and programming simplicity compared to Hadoop.

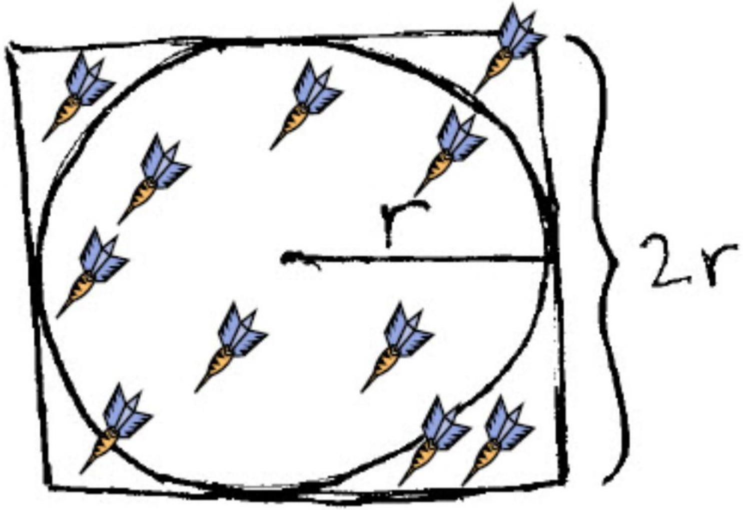
Introduction cont.

Significance

- By comparing the two approaches, we will gain a deeper understanding of:
  - The strengths and limitations of Hadoop MapReduce and PySpark in computational tasks.
  - Practical insights into setting up and running distributed computations.
  - The nuances of performance, scalability, and ease of implementation in both frameworks.
- This project not only provides a hands-on experience with Hadoop and PySpark but also lays the groundwork for understanding their applications in more complex data processing and analytical tasks in the real world.


Circumference
Diameter
Radius
$\frac{\text{Circumference}}{\text{Diameter}} = \pi = 3.14159...$

Design



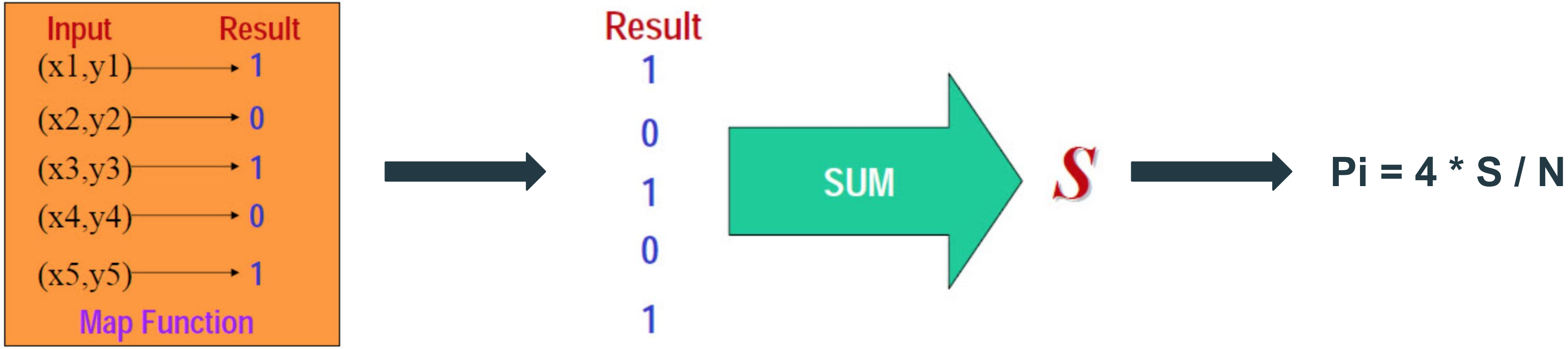
Pi Concept:

- To estimate the value of Pi, we will use a Monte Carlo method. In this approach, we will throw ‘N’ darts at a square dartboard. Each dart lands at a random position (x, y) within the square.
- To determine whether a dart has landed inside the circle inscribed within the square, we use the equation of a circle. Specifically, a dart is inside the circle if the condition `x^2 + y^2 < r^2` is satisfied, where ‘r’ is the radius of the circle.
- By calculating the ratio of darts that land inside the circle to the total number of darts thrown, we can approximate the value of Pi.

Design

Calculating Pi: Process Design

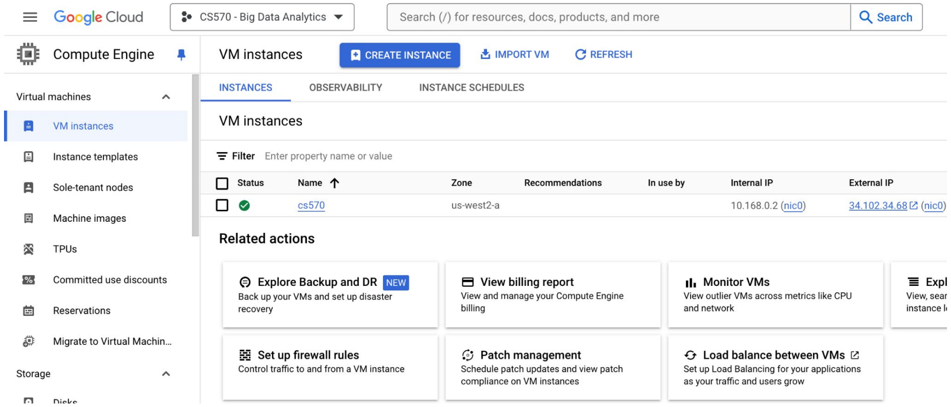
- Generate random coordinates (x, y) within a square.
- Map each (x, y) pair to a result: determine if the point lies inside the circle (assign 1) or outside (assign 0).
- Count the number of points that fall inside the circle.
- Sum up the values to determine the proportion of points inside the circle, which will be used to approximate the value of Pi.



Pi = 4 * S / N

hadoop
Map Reduce

Implementation - Environment



VM instances

□	Status	Name ↑	Zone	Recommendations	In use by	Internal IP	External IP
□	✓	cs570	us-west2-a			10.168.0.2 (nic0)	34.102.34.68 (nic0)

Related actions

Explore Backup and DR NEW
Back up your VMs and set up disaster recovery

View billing report
View and manage your Compute Engine billing

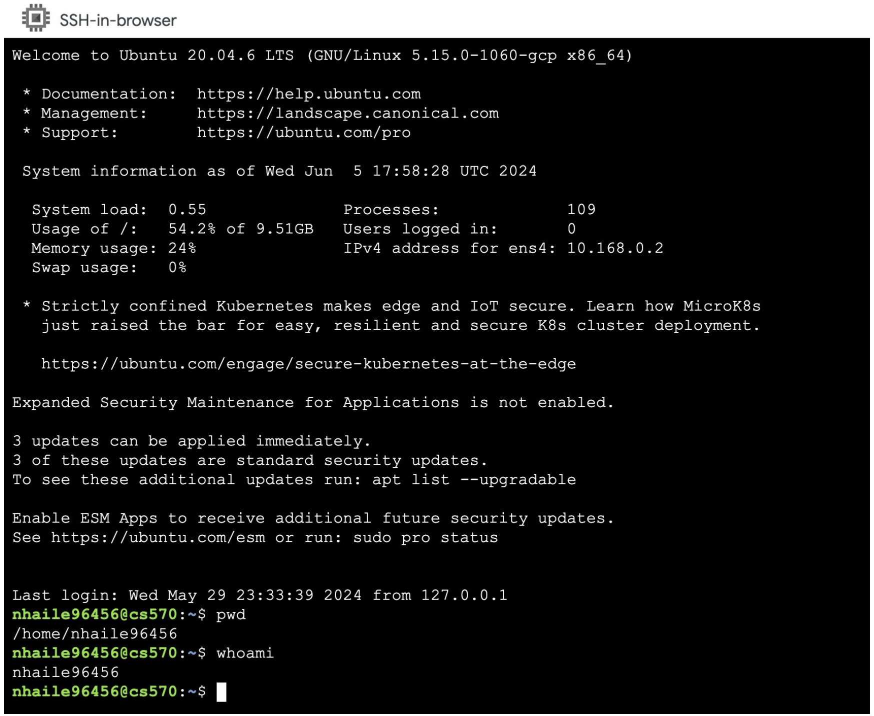
Monitor VMs
View outlier VMs across metrics like CPU and network

Expl
View, sear
instance l

Set up firewall rules
Control traffic to and from a VM instance

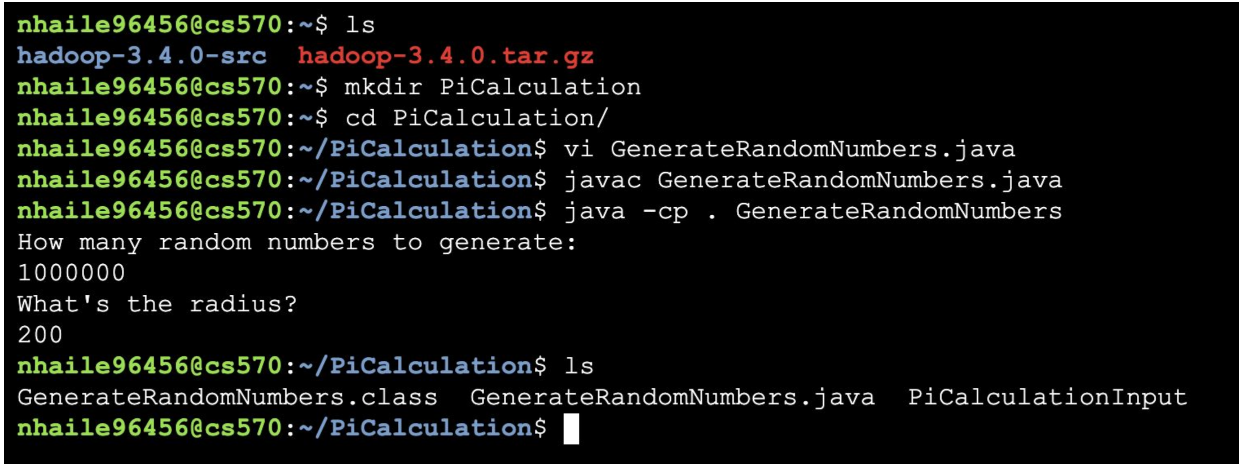
Patch management
Schedule patch updates and view patch compliance on VM instances

Load balance between VMs
Set up Load Balancing for your applications as your traffic and users grow

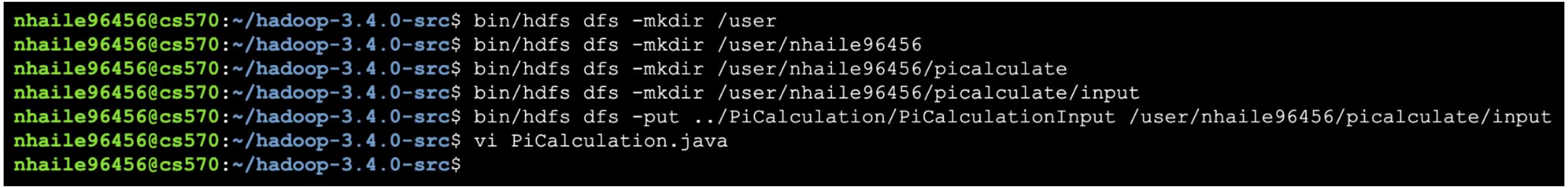


```
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.15.0-1060-gcp x86_64)

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

 System information as of Wed Jun  5 17:58:28 UTC 2024

  System load:  0.55              Processes:             109
  Usage of /:   54.2% of 9.51GB   Users logged in:       0
  Memory usage: 24%               IPv4 address for ens4: 10.168.0.2
  Swap usage:   0%

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

   https://ubuntu.com/engage/secure-kubernetes-at-the-edge

Expanded Security Maintenance for Applications is not enabled.

3 updates can be applied immediately.
3 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

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


Last login: Wed May 29 23:33:39 2024 from 127.0.0.1
nhaile96456@cs570:~$ pwd
/home/nhaile96456
nhaile96456@cs570:~$ whoami
nhaile96456
nhaile96456@cs570:~$
```

Implementation - Prepare Input Data

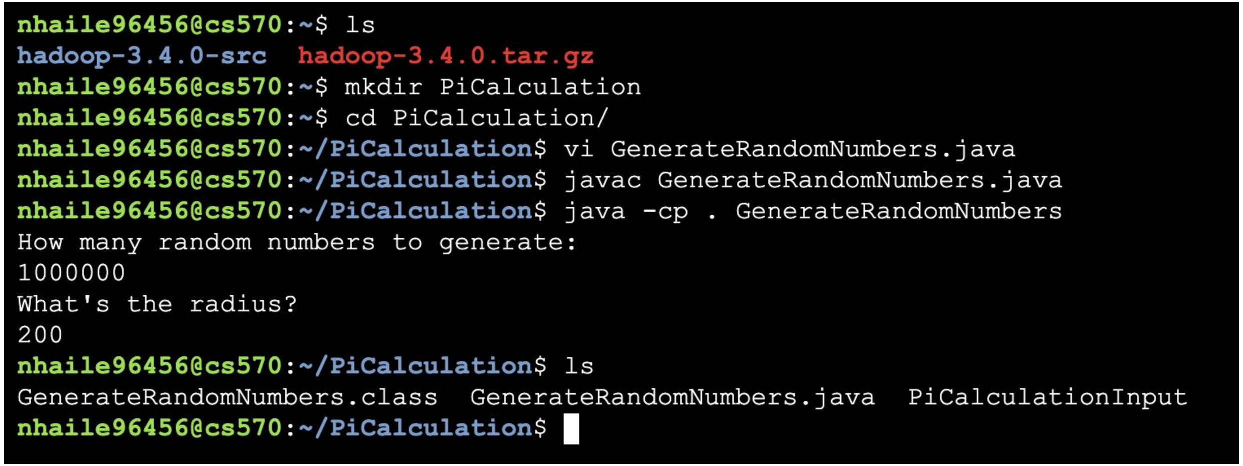


```
nhaile96456@cs570:~$ ls
hadoop-3.4.0-src  hadoop-3.4.0.tar.gz
nhaile96456@cs570:~$ mkdir PiCalculation
nhaile96456@cs570:~$ cd PiCalculation/
nhaile96456@cs570:~/PiCalculation$ vi GenerateRandomNumbers.java
nhaile96456@cs570:~/PiCalculation$ javac GenerateRandomNumbers.java
nhaile96456@cs570:~/PiCalculation$ java -cp . GenerateRandomNumbers
How many random numbers to generate:
1000000
What's the radius?
200
nhaile96456@cs570:~/PiCalculation$ ls
GenerateRandomNumbers.class  GenerateRandomNumbers.java  PiCalculationInput
nhaile96456@cs570:~/PiCalculation$
```

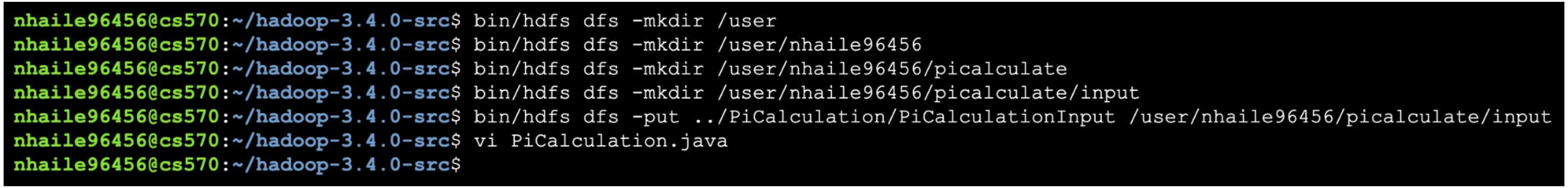


```
nhaile96456@cs570:~/hadoop-3.4.0-src$ bin/hdfs dfs -mkdir /user
nhaile96456@cs570:~/hadoop-3.4.0-src$ bin/hdfs dfs -mkdir /user/nhaile96456
nhaile96456@cs570:~/hadoop-3.4.0-src$ bin/hdfs dfs -mkdir /user/nhaile96456/picalculate
nhaile96456@cs570:~/hadoop-3.4.0-src$ bin/hdfs dfs -mkdir /user/nhaile96456/picalculate/input
nhaile96456@cs570:~/hadoop-3.4.0-src$ bin/hdfs dfs -put ../PiCalculation/PiCalculationInput /user/nhaile96456/picalculate/input
nhaile96456@cs570:~/hadoop-3.4.0-src$ vi PiCalculation.java
nhaile96456@cs570:~/hadoop-3.4.0-src$
```

Implementation - Prepare Input Data

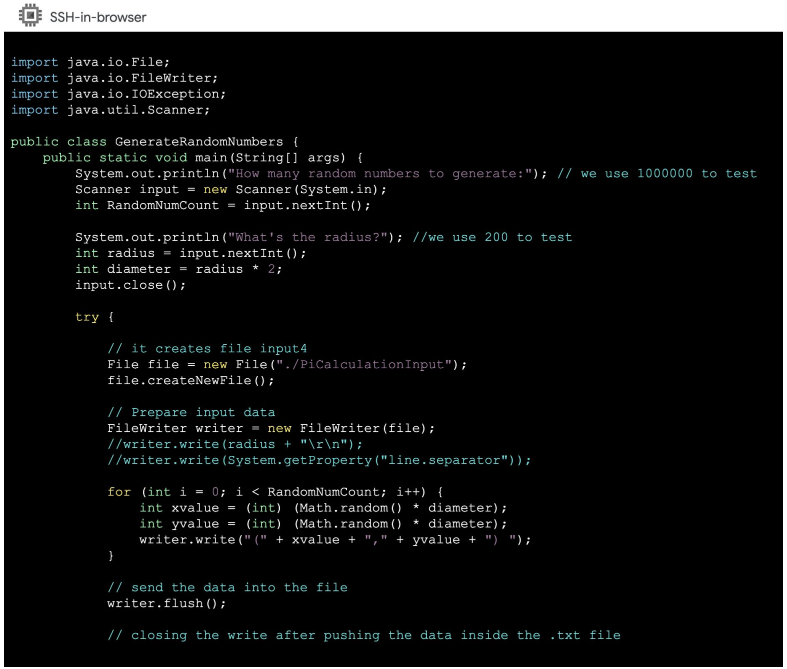


```
nhaile96456@cs570:~$ ls
hadoop-3.4.0-src  hadoop-3.4.0.tar.gz
nhaile96456@cs570:~$ mkdir PiCalculation
nhaile96456@cs570:~$ cd PiCalculation/
nhaile96456@cs570:~/PiCalculation$ vi GenerateRandomNumbers.java
nhaile96456@cs570:~/PiCalculation$ javac GenerateRandomNumbers.java
nhaile96456@cs570:~/PiCalculation$ java -cp . GenerateRandomNumbers
How many random numbers to generate:
1000000
What's the radius?
200
nhaile96456@cs570:~/PiCalculation$ ls
GenerateRandomNumbers.class  GenerateRandomNumbers.java  PiCalculationInput
nhaile96456@cs570:~/PiCalculation$ 
```

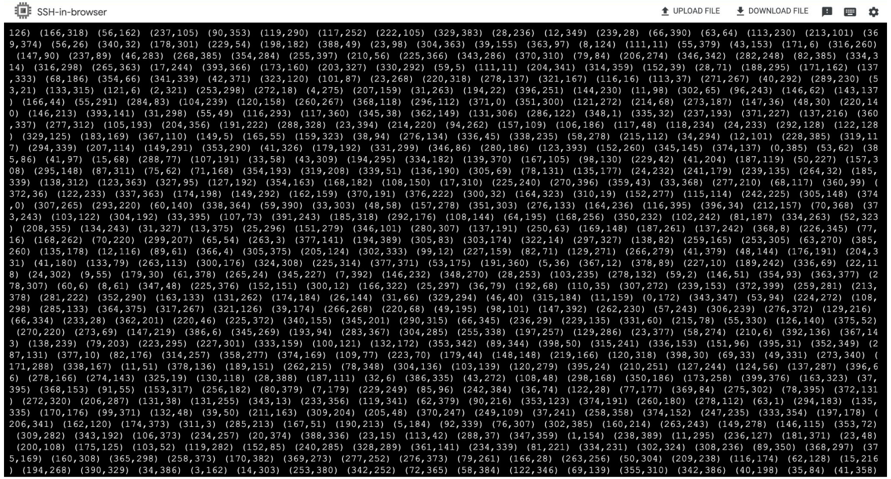


```
nhaile96456@cs570:~/hadoop-3.4.0-src$ bin/hdfs dfs -mkdir /user
nhaile96456@cs570:~/hadoop-3.4.0-src$ bin/hdfs dfs -mkdir /user/nhaile96456
nhaile96456@cs570:~/hadoop-3.4.0-src$ bin/hdfs dfs -mkdir /user/nhaile96456/picalculate
nhaile96456@cs570:~/hadoop-3.4.0-src$ bin/hdfs dfs -mkdir /user/nhaile96456/picalculate/input
nhaile96456@cs570:~/hadoop-3.4.0-src$ bin/hdfs dfs -put ../PiCalculation/PiCalculationInput /user/nhaile96456/picalculate/input
nhaile96456@cs570:~/hadoop-3.4.0-src$ vi PiCalculation.java
nhaile96456@cs570:~/hadoop-3.4.0-src$
```

Implementation - Prepare Input Data

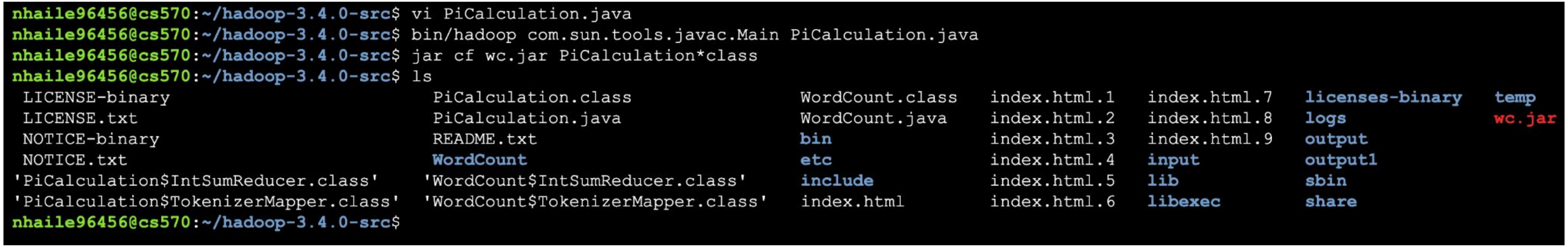


```java
import java.io.File;
import java.io.FileWriter;
import java.io.IOException;
import java.util.Scanner;

public class GenerateRandomNumbers {
    public static void main(String[] args) {
        System.out.println("How many random numbers to generate:"); // we use 1000000 to test
        Scanner input = new Scanner(System.in);
        int RandomNumCount = input.nextInt();

        System.out.println("What's the radius?"); //we use 200 to test
        int radius = input.nextInt();
        int diameter = radius * 2;
        input.close();

        try {

            // it creates file input4
            File file = new File("./PiCalculationInput");
            file.createNewFile();

            // Prepare input data
            FileWriter writer = new FileWriter(file);
            //writer.write(radius + "\r\n");
            //writer.write(System.getProperty("line.separator"));

            for (int i = 0; i < RandomNumCount; i++) {
                int xvalue = (int) (Math.random() * diameter);
                int yvalue = (int) (Math.random() * diameter);
                writer.write("(" + xvalue + "," + yvalue + ") ");
            }

            // send the data into the file
            writer.flush();

            // closing the write after pushing the data inside the .txt file
```

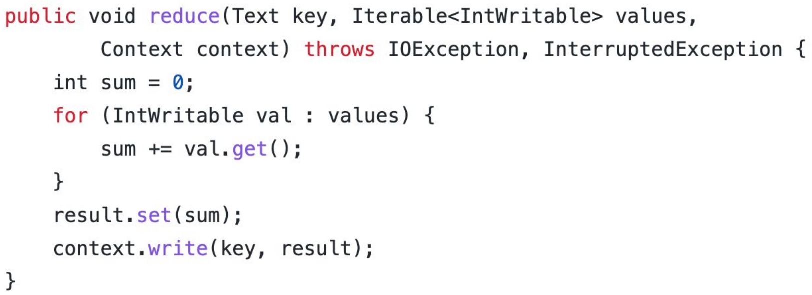


```
126) (166,318) (56,162) (237,105) (90,353) (119,290) (117,252) (222,105) (329,383) (28,236) (12,349) (239,28) (66,390) (63,64) (113,230) (213,101) (36
9,374) (56,26) (340,32) (178,301) (229,54) (198,182) (388,49) (23,98) (304,363) (39,155) (363,97) (8,124) (111,11) (55,379) (43,153) (171,6) (316,260)
 (147,90) (237,89) (46,283) (268,385) (354,284) (255,397) (210,56) (225,366) (343,286) (370,310) (79,84) (206,274) (346,342) (282,248) (82,385) (334,3
14) (316,298) (265,363) (17,244) (393,366) (173,160) (203,327) (330,292) (59,5) (111,11) (204,341) (314,359) (152,39) (28,71) (188,295) (171,162) (137
,333) (68,186) (354,66) (341,339) (42,371) (323,120) (101,87) (23,268) (220,318) (278,137) (321,167) (116,16) (113,37) (271,267) (40,292) (289,230) (5
3,21) (133,315) (121,6) (2,321) (253,298) (272,18) (4,275) (207,159) (31,263) (194,22) (396,251) (144,230) (11,98) (302,65) (96,243) (146,62) (143,137
) (166,44) (55,291) (284,83) (104,239) (120,158) (260,267) (368,118) (296,112) (371,0) (351,300) (121,272) (214,68) (273,187) (147,36) (48,30) (220,14
0) (146,213) (393,141) (31,298) (55,49) (116,293) (117,360) (345,38) (362,149) (131,306) (286,122) (348,1) (335,32) (237,193) (371,227) (137,216) (360
,337) (277,312) (105,193) (204,356) (191,222) (288,328) (23,394) (214,220) (94,262) (157,109) (106,186) (117,48) (118,234) (24,233) (292,128) (122,128
) (329,125) (183,169) (367,110) (149,5) (165,55) (159,323) (38,94) (276,134) (336,45) (338,235) (58,278) (215,112) (34,294) (12,101) (228,385) (319,11
7) (294,339) (207,114) (149,291) (353,290) (41,326) (179,192) (331,299) (346,86) (280,186) (123,393) (152,260) (345,145) (374,137) (0,385) (53,62) (38
5,86) (41,97) (15,68) (288,77) (107,191) (33,58) (43,309) (194,295) (334,182) (139,370) (167,105) (98,130) (229,42) (41,204) (187,119) (50,227) (157,3
08) (295,148) (87,311) (75,62) (71,168) (354,193) (319,208) (339,51) (136,190) (305,69) (78,131) (135,177) (24,232) (241,179) (239,135) (264,32) (185,
339) (138,312) (123,363) (327,95) (127,192) (354,163) (168,182) (108,150) (17,310) (225,240) (270,396) (359,43) (33,368) (277,210) (68,117) (360,99) (
372,36) (122,233) (337,363) (174,198) (149,292) (162,159) (370,191) (376,222) (300,32) (164,323) (310,19) (152,277) (115,114) (242,225) (305,148) (374
,0) (307,265) (293,220) (60,140) (338,364) (59,390) (33,303) (48,58) (157,278) (351,303) (276,133) (164,236) (116,395) (396,34) (212,157) (70,368) (37
3,243) (103,122) (304,192) (33,395) (107,73) (391,243) (185,318) (292,176) (108,144) (64,195) (168,256) (350,232) (102,242) (81,187) (334,263) (52,323
) (208,355) (134,243) (31,327) (13,375) (25,296) (151,279) (346,101) (280,307) (137,191) (250,63) (169,148) (187,261) (137,242) (368,8) (226,345) (77,
16) (168,262) (70,220) (299,207) (65,54) (263,3) (377,141) (194,389) (305,83) (303,174) (322,14) (297,327) (138,82) (259,165) (253,305) (63,270) (385,
260) (135,178) (12,116) (89,61) (366,4) (305,375) (205,124) (302,333) (99,12) (227,159) (82,71) (129,271) (266,279) (41,379) (48,144) (176,191) (204,3
31) (41,180) (133,79) (263,113) (300,176) (324,308) (225,314) (377,371) (53,175) (191,360) (5,36) (367,12) (378,89) (227,10) (189,242) (336,69) (22,11
8) (24,302) (9,55) (179,30) (61,378) (265,24) (345,227) (7,392) (146,232) (348,270) (28,253) (103,235) (278,132) (59,2) (146,51) (354,93) (363,377) (2
78,307) (60,6) (8,61) (347,48) (225,376) (152,151) (300,12) (166,322) (25,297) (36,79) (192,68) (110,35) (307,272) (239,153) (372,399) (259,281) (213,
378) (281,222) (352,290) (163,133) (131,262) (174,184) (26,144) (31,66) (329,294) (46,40) (315,184) (11,159) (0,172) (343,347) (53,94) (224,272) (108,
298) (285,133) (364,375) (317,267) (321,126) (39,174) (266,268) (220,68) (49,195) (98,101) (147,392) (262,230) (57,243) (306,239) (276,372) (129,216)
(66,334) (233,28) (362,201) (220,46) (225,372) (340,155) (345,201) (290,315) (66,345) (236,29) (229,135) (331,60) (215,78) (55,330) (126,140) (375,52)
 (270,220) (273,69) (147,219) (386,6) (345,269) (193,94) (283,367) (304,285) (255,338) (197,257) (129,286) (23,377) (58,274) (210,6) (392,136) (367,14
3) (138,239) (79,203) (223,295) (227,301) (333,159) (100,121) (132,172) (353,342) (89,344) (398,50) (315,241) (336,153) (151,96) (395,31) (352,349) (2
87,131) (377,10) (82,176) (314,257) (358,277) (374,169) (109,77) (223,70) (179,44) (148,148) (219,166) (120,318) (398,30) (69,33) (49,331) (273,340) (
171,288) (338,167) (11,51) (378,136) (189,151) (262,215) (78,348) (304,136) (103,139) (120,279) (395,24) (210,251) (127,244) (124,56) (137,287) (396,6
6) (278,166) (274,143) (325,19) (130,118) (28,388) (187,111) (32,6) (386,335) (43,272) (108,48) (298,168) (350,186) (173,258) (399,376) (163,323) (37,
395) (368,153) (91,55) (153,317) (256,182) (80,379) (7,179) (229,249) (85,96) (242,384) (36,74) (122,28) (77,177) (369,84) (275,302) (78,395) (372,131
) (272,320) (206,287) (131,38) (131,255) (343,13) (233,356) (119,341) (62,379) (90,216) (353,123) (374,191) (260,180) (278,112) (63,1) (294,183) (135,
335) (170,176) (99,371) (132,48) (39,50) (211,163) (309,204) (205,48) (370,247) (249,109) (37,241) (258,358) (374,152) (247,235) (333,354) (197,178) (
206,341) (162,120) (174,373) (311,3) (285,213) (167,51) (190,213) (5,184) (92,339) (76,307) (302,385) (160,214) (263,243) (149,278) (146,115) (353,72)
 (309,282) (343,192) (106,373) (234,257) (20,374) (388,336) (23,15) (113,42) (288,37) (347,359) (1,154) (238,389) (11,295) (236,127) (181,371) (23,48)
 (200,108) (175,125) (103,52) (119,282) (152,85) (240,285) (328,289) (361,141) (234,339) (81,221) (334,231) (302,324) (308,236) (89,350) (368,297) (37
5,169) (160,308) (365,298) (258,373) (170,382) (369,273) (277,252) (276,373) (79,261) (166,28) (263,256) (50,304) (209,238) (116,174) (62,128) (15,216
) (194,268) (390,329) (34,386) (3,162) (14,303) (253,380) (342,252) (72,365) (58,384) (122,346) (69,139) (355,310) (342,386) (40,198) (35,84) (41,358)
```

Implementation - MapReduce Program



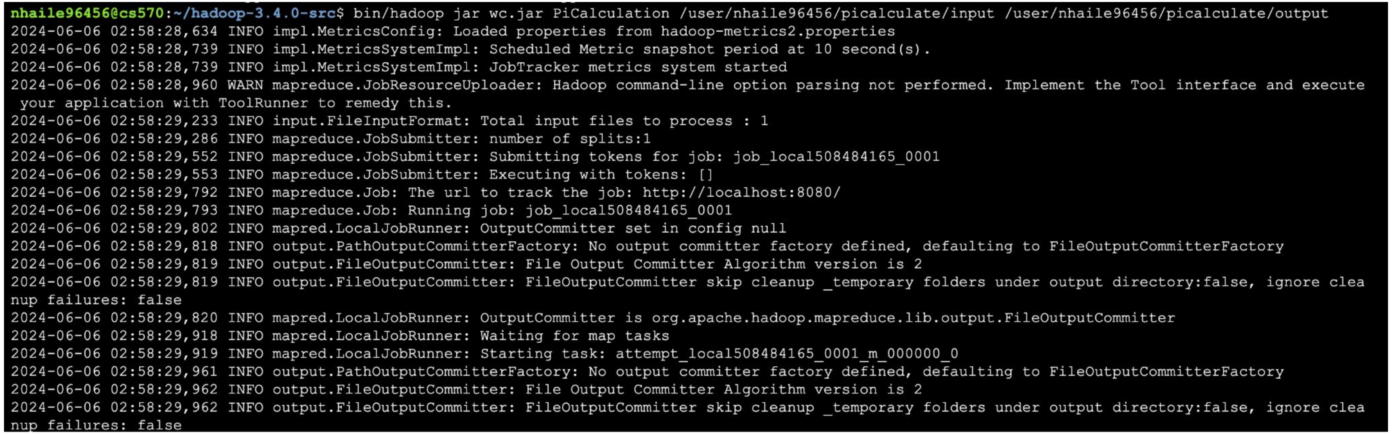
```
nhaile96456@cs570:~/hadoop-3.4.0-src$ vi PiCalculation.java
nhaile96456@cs570:~/hadoop-3.4.0-src$ bin/hadoop com.sun.tools.javac.Main PiCalculation.java
nhaile96456@cs570:~/hadoop-3.4.0-src$ jar cf wc.jar PiCalculation*class
nhaile96456@cs570:~/hadoop-3.4.0-src$ ls
 LICENSE-binary                         PiCalculation.class                 WordCount.class   index.html.1  index.html.7  licenses-binary  temp
 LICENSE.txt                            PiCalculation.java                  WordCount.java    index.html.2  index.html.8  logs             wc.jar
 NOTICE-binary                          README.txt                          bin               index.html.3  index.html.9  output
 NOTICE.txt                             WordCount                           etc               index.html.4  input         output1
'PiCalculation$IntSumReducer.class'    'WordCount$IntSumReducer.class'      include           index.html.5  lib           sbin
'PiCalculation$TokenizerMapper.class'  'WordCount$TokenizerMapper.class'    index.html        index.html.6  libexec       share
nhaile96456@cs570:~/hadoop-3.4.0-src$
```



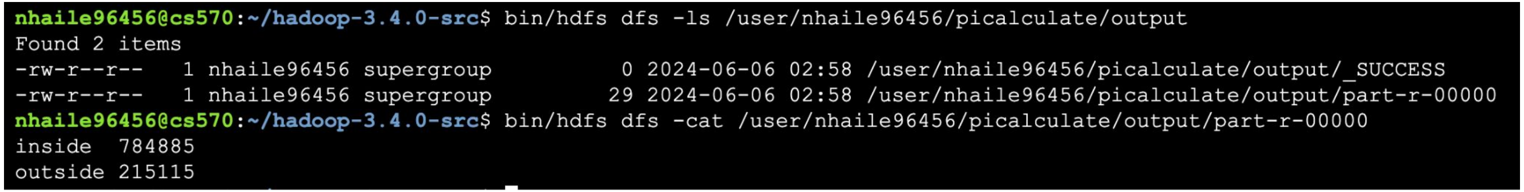
```java
public void reduce(Text key, Iterable<IntWritable> values,
        Context context) throws IOException, InterruptedException {
    int sum = 0;
    for (IntWritable val : values) {
        sum += val.get();
    }
    result.set(sum);
    context.write(key, result);
}
```

Implementation - Execution

$ bin/hadoop jar wc.jar PiCalculation /user/nhaile96456/picalculate/input /user/nhaile96456/picalculate/output
$ bin/hdfs dfs -ls /user/nhaile96456/picalculate/output
$ bin/hdfs dfs -cat /user/nhaile96456/picalculate/output/part-r-00000



```
nhaile96456@cs570:~/hadoop-3.4.0-src$ bin/hadoop jar wc.jar PiCalculation /user/nhaile96456/picalculate/input /user/nhaile96456/picalculate/output
2024-06-06 02:58:28,634 INFO impl.MetricsConfig: Loaded properties from hadoop-metrics2.properties
2024-06-06 02:58:28,739 INFO impl.MetricsSystemImpl: Scheduled Metric snapshot period at 10 second(s).
2024-06-06 02:58:28,739 INFO impl.MetricsSystemImpl: JobTracker metrics system started
2024-06-06 02:58:28,960 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute
 your application with ToolRunner to remedy this.
2024-06-06 02:58:29,233 INFO input.FileInputFormat: Total input files to process : 1
2024-06-06 02:58:29,286 INFO mapreduce.JobSubmitter: number of splits:1
2024-06-06 02:58:29,552 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_local508484165_0001
2024-06-06 02:58:29,553 INFO mapreduce.JobSubmitter: Executing with tokens: []
2024-06-06 02:58:29,792 INFO mapreduce.Job: The url to track the job: http://localhost:8080/
2024-06-06 02:58:29,793 INFO mapreduce.Job: Running job: job_local508484165_0001
2024-06-06 02:58:29,802 INFO mapred.LocalJobRunner: OutputCommitter set in config null
2024-06-06 02:58:29,818 INFO output.PathOutputCommitterFactory: No output committer factory defined, defaulting to FileOutputCommitterFactory
2024-06-06 02:58:29,819 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 2
2024-06-06 02:58:29,819 INFO output.FileOutputCommitter: FileOutputCommitter skip cleanup _temporary folders under output directory:false, ignore clea
nup failures: false
2024-06-06 02:58:29,820 INFO mapred.LocalJobRunner: OutputCommitter is org.apache.hadoop.mapreduce.lib.output.FileOutputCommitter
2024-06-06 02:58:29,918 INFO mapred.LocalJobRunner: Waiting for map tasks
2024-06-06 02:58:29,919 INFO mapred.LocalJobRunner: Starting task: attempt_local508484165_0001_m_000000_0
2024-06-06 02:58:29,961 INFO output.PathOutputCommitterFactory: No output committer factory defined, defaulting to FileOutputCommitterFactory
2024-06-06 02:58:29,962 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 2
2024-06-06 02:58:29,962 INFO output.FileOutputCommitter: FileOutputCommitter skip cleanup _temporary folders under output directory:false, ignore clea
nup failures: false
```



```
nhaile96456@cs570:~/hadoop-3.4.0-src$ bin/hdfs dfs -ls /user/nhaile96456/picalculate/output
Found 2 items
-rw-r--r--   1 nhaile96456 supergroup          0 2024-06-06 02:58 /user/nhaile96456/picalculate/output/_SUCCESS
-rw-r--r--   1 nhaile96456 supergroup         29 2024-06-06 02:58 /user/nhaile96456/picalculate/output/part-r-00000
nhaile96456@cs570:~/hadoop-3.4.0-src$ bin/hdfs dfs -cat /user/nhaile96456/picalculate/output/part-r-00000
inside  784885
outside 215115
```

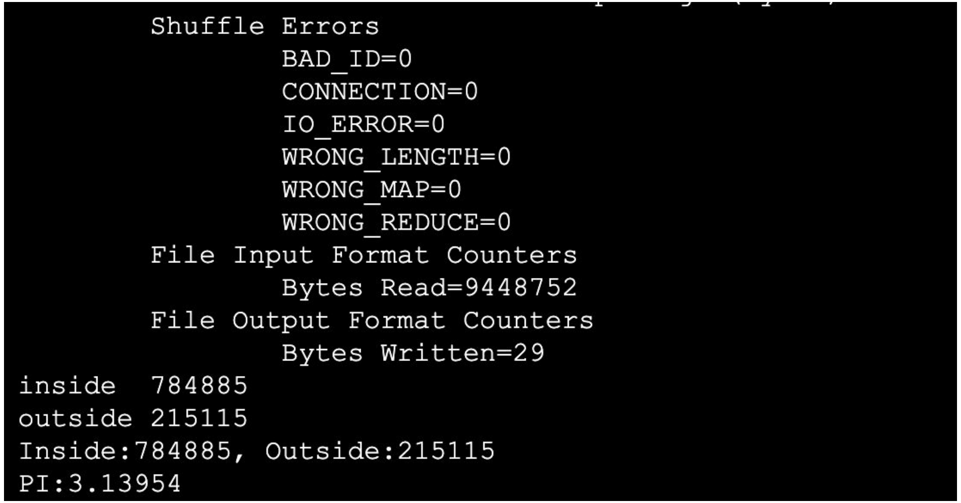
Test

Result:

S = 784885

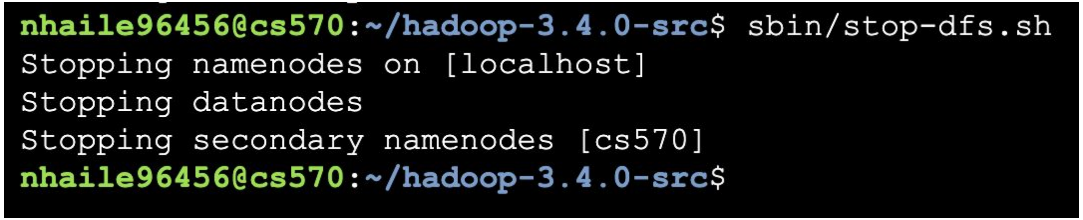
N = 1000000

Pi = 4 * S / N = 4 * 784885 / 1000000 = 3.13954

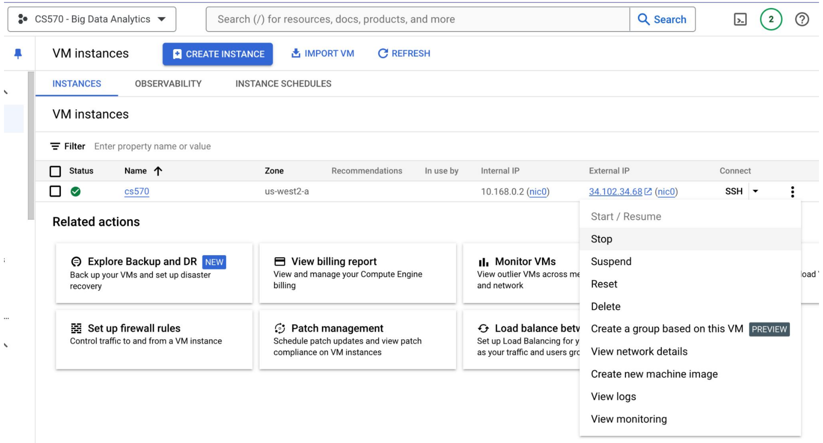


```
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=9448752
        File Output Format Counters
                Bytes Written=29
inside  784885
outside 215115
Inside:784885, Outside:215115
PI:3.13954
```

Test cont.



```
nhaile96456@cs570:~/hadoop-3.4.0-src$ sbin/stop-dfs.sh
Stopping namenodes on [localhost]
Stopping datanodes
Stopping secondary namenodes [cs570]
nhaile96456@cs570:~/hadoop-3.4.0-src$
```



VM instances

Filter Enter property name or value

□	Status	Name ↑	Zone	Recommendations	In use by	Internal IP	External IP	Connect
□	✓	cs570	us-west2-a			10.168.0.2 (nic0)	34.102.34.68 (nic0)	SSH

Related actions

Explore Backup and DR NEW
Back up your VMs and set up disaster recovery

View billing report
View and manage your Compute Engine billing

Monitor VMs
View outlier VMs across me and network

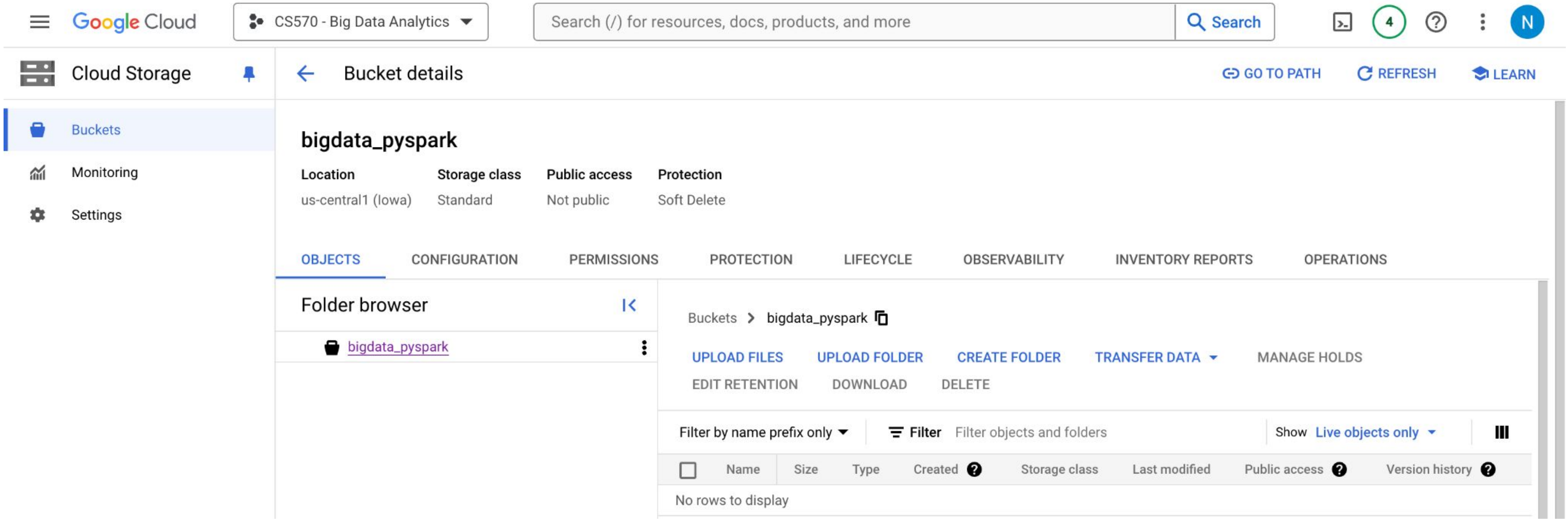
Set up firewall rules
Control traffic to and from a VM instance

Patch management
Schedule patch updates and view patch compliance on VM instances

Load balance betv
Set up Load Balancing for y as your traffic and users grc

APACHE
PySpark
TM

Implementation - Environment



Google Cloud

CS570 - Big Data Analytics

Search (/) for resources, docs, products, and more

Search

Cloud Storage

- Buckets
- Monitoring
- Settings

Bucket details

GO TO PATH REFRESH LEARN

bigdata_pyspark

Location	Storage class	Public access	Protection
us-central1 (Iowa)	Standard	Not public	Soft Delete

OBJECTS CONFIGURATION PERMISSIONS PROTECTION LIFECYCLE OBSERVABILITY INVENTORY REPORTS OPERATIONS

Folder browser

bigdata_pyspark

Buckets > bigdata_pyspark

UPLOAD FILES UPLOAD FOLDER CREATE FOLDER TRANSFER DATA MANAGE HOLDS

EDIT RETENTION DOWNLOAD DELETE

Filter by name prefix only

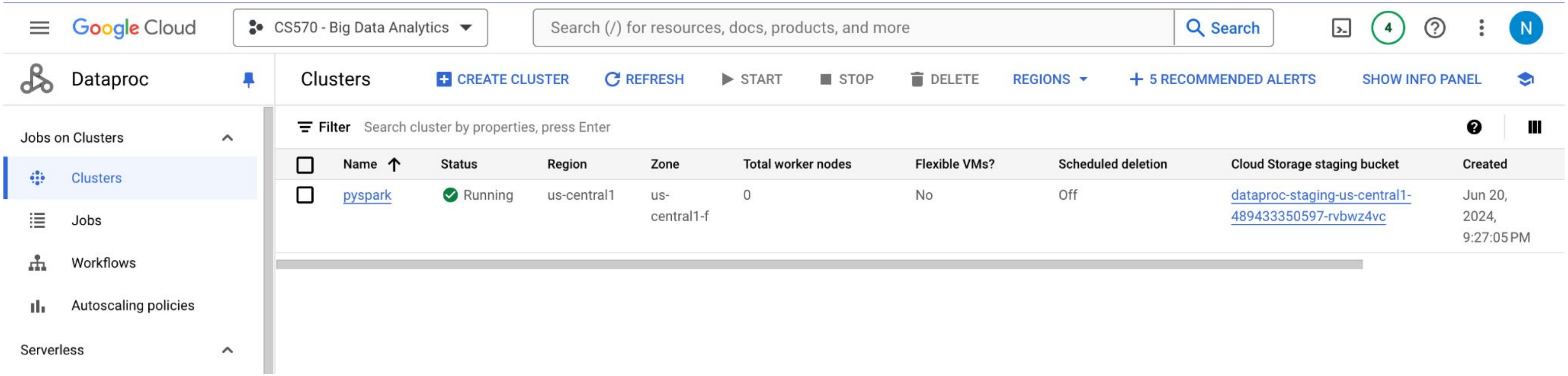
Filter Filter objects and folders

Show Live objects only

□	Name	Size	Type	Created	Storage class	Last modified	Public access	Version history

No rows to display

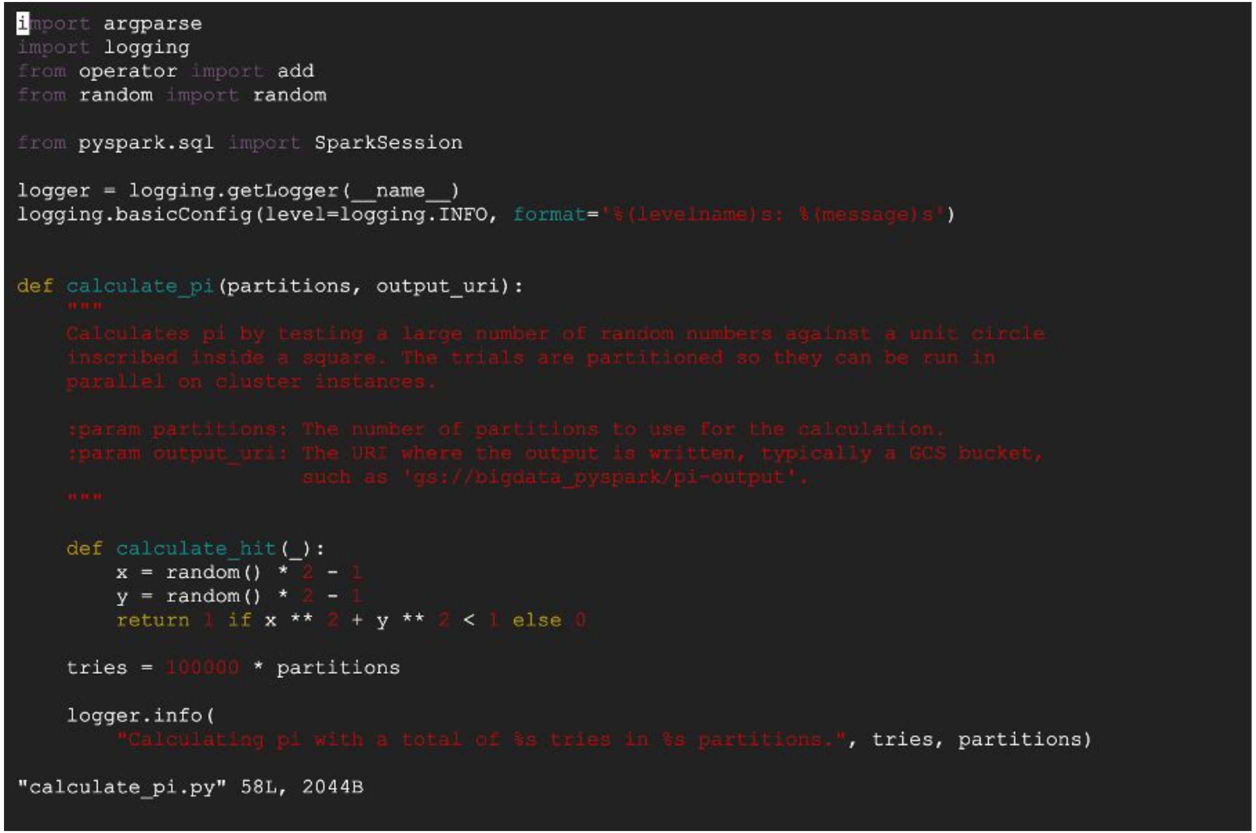
Implementation - Environment



☐	Name ↑	Status	Region	Zone	Total worker nodes	Flexible VMs?	Scheduled deletion	Cloud Storage staging bucket	Created
☐	pyspark	✓ Running	us-central1	us-central1-f	0	No	Off	dataproc-staging-us-central1-489433350597-rvbwz4vc	Jun 20, 2024, 9:27:05 PM

Implementation - Prepare the Input File

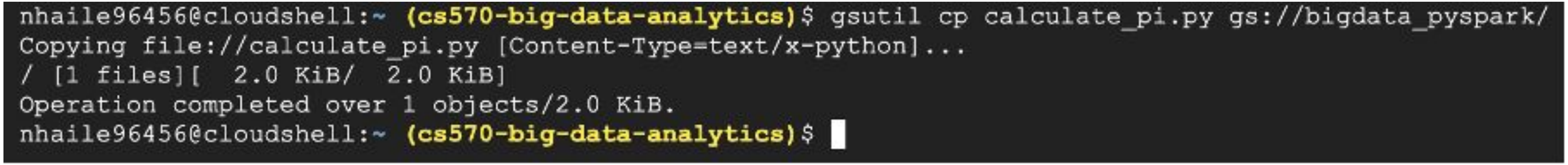
In the cloud shell create the calculate_py.py script and copy it to the cloud bucket storage



```
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 a GCS bucket,
                       such as 'gs://bigdata_pyspark/pi-output'.
    """

    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)

"calculate_pi.py" 58L, 2044B
```

Implementation - Prepare the Input File

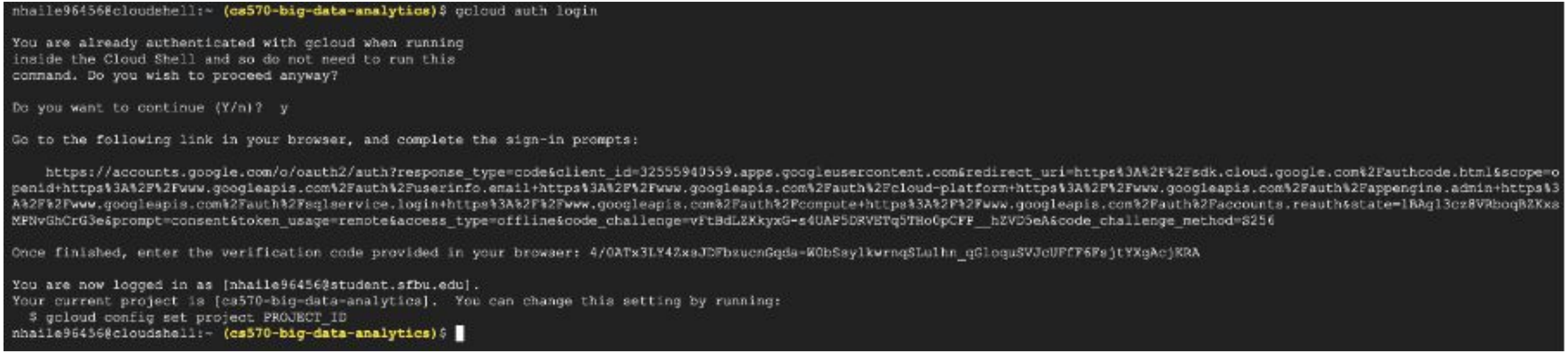
Upload the Script to Cloud Storage bucket for easy access, i.e., your newly created bucket.



```
nhaile96456@cloudshell:~ (cs570-big-data-analytics)$ gsutil cp calculate_pi.py gs://bigdata_pyspark/
Copying file://calculate_pi.py [Content-Type=text/x-python]...
/ [1 files][  2.0 KiB/  2.0 KiB]
Operation completed over 1 objects/2.0 KiB.
nhaile96456@cloudshell:~ (cs570-big-data-analytics)$
```

Implementation - Running the Script

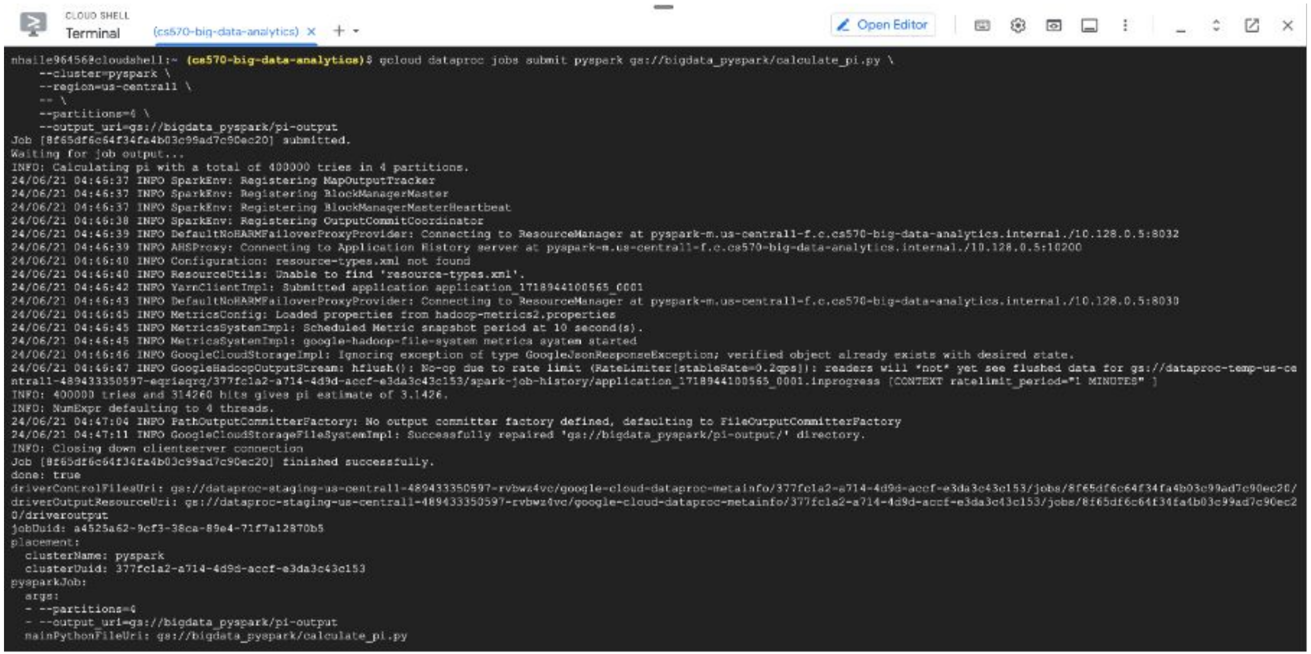
Login to Google Cloud through the CLI.



```
nhaile96456@cloudshell:~ (cs570-big-data-analytics)$ gcloud auth login

You are already authenticated with gcloud when running
inside the Cloud Shell and so do not need to run this
command. Do you wish to proceed anyway?

Do you want to continue (Y/n)?  y

Go to the following link in your browser, and complete the sign-in prompts:

    https://accounts.google.com/o/oauth2/auth?response_type=code&client_id=32555940559.apps.googleusercontent.com&redirect_uri=https%3A%2F%2Fsdk.cloud.google.com%2Fauthcode.html&scope=o
penid+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fuserinfo.email+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcloud-platform+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fappengine.admin+https%3
A%2F%2Fwww.googleapis.com%2Fauth%2Fsqlservice.login+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcompute+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Faccounts.reauth&state=1BAgl3cz8VRboqBZKxs
MPNvGhCrG3e&prompt=consent&token_usage=remote&access_type=offline&code_challenge=vFtBdLZKkyxG-s4UAP5DRVETq5THo0pCFF__hZVD5eA&code_challenge_method=S256

Once finished, enter the verification code provided in your browser: 4/0ATx3LY4ZxsJDFbzucnGqda-WObSsylkwrnqSLulhn_qGloquSVJcUFfF6FsjtYXgAcjKRA

You are now logged in as [nhaile96456@student.sfbu.edu].
Your current project is [cs570-big-data-analytics].  You can change this setting by running:
  $ gcloud config set project PROJECT_ID
nhaile96456@cloudshell:~ (cs570-big-data-analytics)$ █
```

Implementation - Running the Script

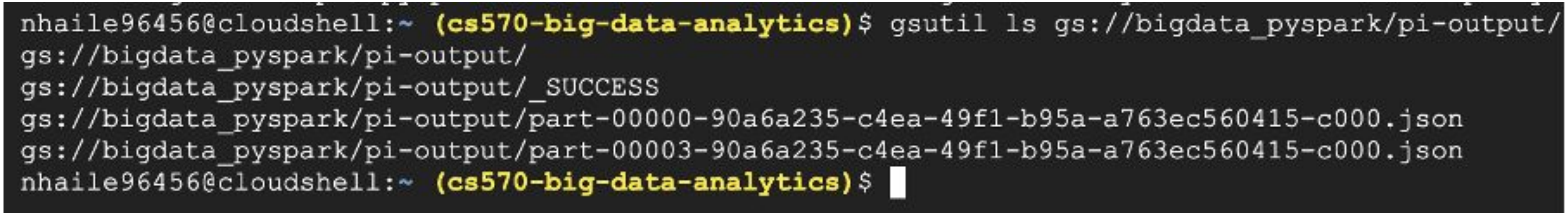
Submitting the PySpark job to Dataproc.



```
mhaile96456@cloudshell:~ (cs570-big-data-analytics)$ gcloud dataproc jobs submit pyspark gs://bigdata_pyspark/calculate_pi.py \
    --cluster=pyspark \
    --region=us-central1 \
    -- \
    --partitions=4 \
    --output_uri=gs://bigdata_pyspark/pi-output
Job [8f65df6c64f34fa4b03c99ad7c90ec20] submitted.
Waiting for job output...
INFO: Calculating pi with a total of 400000 tries in 4 partitions.
24/06/21 04:46:37 INFO SparkEnv: Registering MapOutputTracker
24/06/21 04:46:37 INFO SparkEnv: Registering BlockManagerMaster
24/06/21 04:46:37 INFO SparkEnv: Registering BlockManagerMasterHeartbeat
24/06/21 04:46:38 INFO SparkEnv: Registering OutputCommitCoordinator
24/06/21 04:46:39 INFO DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at pyspark-m.us-central1-f.c.cs570-big-data-analytics.internal./10.128.0.5:8032
24/06/21 04:46:39 INFO AHSProxy: Connecting to Application History server at pyspark-m.us-central1-f.c.cs570-big-data-analytics.internal./10.128.0.5:10200
24/06/21 04:46:40 INFO Configuration: resource-types.xml not found
24/06/21 04:46:40 INFO ResourceUtils: Unable to find 'resource-types.xml'.
24/06/21 04:46:42 INFO YarnClientImpl: Submitted application application_1718944100565_0001
24/06/21 04:46:43 INFO DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at pyspark-m.us-central1-f.c.cs570-big-data-analytics.internal./10.128.0.5:8030
24/06/21 04:46:45 INFO MetricsConfig: Loaded properties from hadoop-metrics2.properties
24/06/21 04:46:45 INFO MetricsSystemImpl: Scheduled Metric snapshot period at 10 second(s).
24/06/21 04:46:45 INFO MetricsSystemImpl: google-hadoop-file-system metrics system started
24/06/21 04:46:46 INFO GoogleCloudStorageImpl: Ignoring exception of type GoogleJsonResponseException; verified object already exists with desired state.
24/06/21 04:46:47 INFO GoogleHadoopOutputStream: hflush(): No-op due to rate limit (RateLimiter[stableRate=0.2qps]): readers will *not* yet see flushed data for gs://dataproc-temp-us-ce
ntral1-489433350597-eqriaqrq/377fc1a2-a714-4d9d-accf-e3da3c43c153/spark-job-history/application_1718944100565_0001.inprogress [CONTEXT ratelimit_period="1 MINUTES" ]
INFO: 400000 tries and 314260 hits gives pi estimate of 3.1426.
INFO: NumExpr defaulting to 4 threads.
24/06/21 04:47:04 INFO PathOutputCommitterFactory: No output committer factory defined, defaulting to FileOutputCommitterFactory
24/06/21 04:47:11 INFO GoogleCloudStorageFileSystemImpl: Successfully repaired 'gs://bigdata_pyspark/pi-output/' directory.
INFO: Closing down clientserver connection
Job [8f65df6c64f34fa4b03c99ad7c90ec20] finished successfully.
done: true
driverControlFilesUri: gs://dataproc-staging-us-central1-489433350597-rvbwz4vc/google-cloud-dataproc-metainfo/377fc1a2-a714-4d9d-accf-e3da3c43c153/jobs/8f65df6c64f34fa4b03c99ad7c90ec20/
driverOutputResourceUri: gs://dataproc-staging-us-central1-489433350597-rvbwz4vc/google-cloud-dataproc-metainfo/377fc1a2-a714-4d9d-accf-e3da3c43c153/jobs/8f65df6c64f34fa4b03c99ad7c90ec2
0/driveroutput
jobUuid: a4525a62-9cf3-38ca-89e4-71f7a12870b5
placement:
  clusterName: pyspark
  clusterUuid: 377fc1a2-a714-4d9d-accf-e3da3c43c153
pysparkJob:
  args:
  - --partitions=4
  - --output_uri=gs://bigdata_pyspark/pi-output
  mainPythonFileUri: gs://bigdata_pyspark/calculate_pi.py
```

Implementation - Verifying the Output

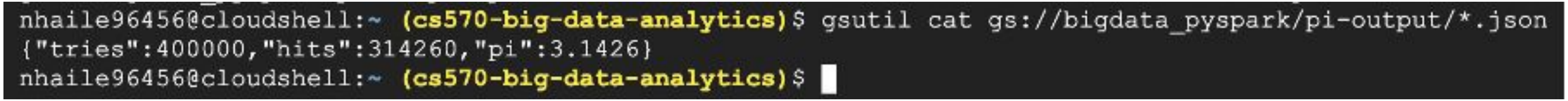
Check the output.



```
nhaile96456@cloudshell:~ (cs570-big-data-analytics)$ gsutil ls gs://bigdata_pyspark/pi-output/
gs://bigdata_pyspark/pi-output/
gs://bigdata_pyspark/pi-output/_SUCCESS
gs://bigdata_pyspark/pi-output/part-00000-90a6a235-c4ea-49f1-b95a-a763ec560415-c000.json
gs://bigdata_pyspark/pi-output/part-00003-90a6a235-c4ea-49f1-b95a-a763ec560415-c000.json
nhaile96456@cloudshell:~ (cs570-big-data-analytics)$
```

Implementation - Verifying the Output

Check the output.



```
nhaile96456@cloudshell:~ (cs570-big-data-analytics)$ gsutil cat gs://bigdata_pyspark/pi-output/*.json
{"tries":400000,"hits":314260,"pi":3.1426}
nhaile96456@cloudshell:~ (cs570-big-data-analytics)$
```

Enhancement Ideas

Fixed Radius, Variable Random Number Size:

- Maintain a constant radius size while varying the quantity of random points generated.
- Analyze how changes in the number of random points affect the accuracy of the Pi calculation.

Fixed Random Number Size, Variable Radius:

- Keep the number of random points constant while altering the radius size.
- Investigate how different radius sizes impact the precision of the Pi estimation.

Accuracy Optimization:

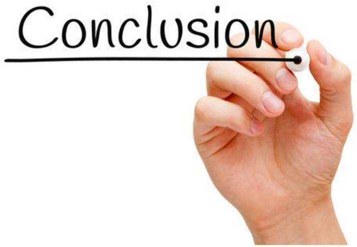
- Conduct comparative experiments to identify the configuration that yields the most accurate approximation of Pi.
- Use these findings to optimize the process for better precision.

Significance

- Gain deeper insights into factors affecting the accuracy of distributed computations, improve other tasks relying on random sampling techniques, and assess the performance of Hadoop MapReduce and PySpark.

Conclusion

- This project provided valuable hands-on experience with both Hadoop MapReduce and PySpark, highlighting their strengths and limitations in distributed computing.
- By comparing these frameworks, we gained a deeper understanding of their practical applications in large-scale data processing and computational tasks.
- This exploration not only enhanced our knowledge of distributed computing but also laid the groundwork for more advanced and precise computational projects in the future.



References

Python - Calculating π number with Apache Spark

Time for action – using Hadoop to calculate Pi

Overview of Pi calculation using MapReduce

Pi Computation With MapReduce

Calculating Pi with Apache Spark

Run a program to estimate pi | Data Science with Apache Spark

GitHub Link

- https://github.com/cur10usityDrives/Big-Data/tree/main/PySpark/Pi

THANK YOU