Creating PySpark program to calculating pi

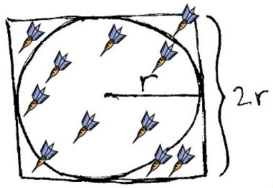
Jisen Fang

Table of Content

Theory

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

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



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

Title

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

Formula:

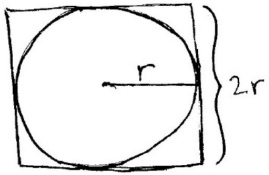
4 * S / N = 4 * (pi * r * r) / (4 * r * r) = pi

Note:

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

Sample MapReduce Code- Estimate π

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



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

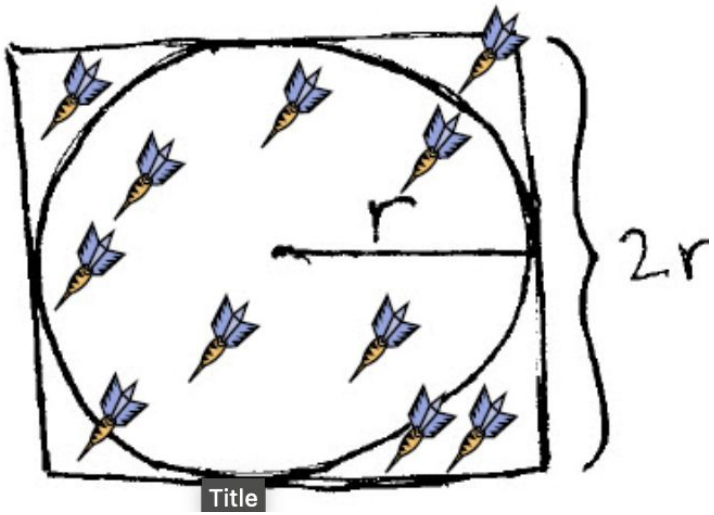
$$\Rightarrow \pi = 4P(dart\ in\ circle)$$

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

How?

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

   Pi = 4(S/N)



Create input files

Randomly generate 200 coordinates in (x,y) format and write them into the input.txt

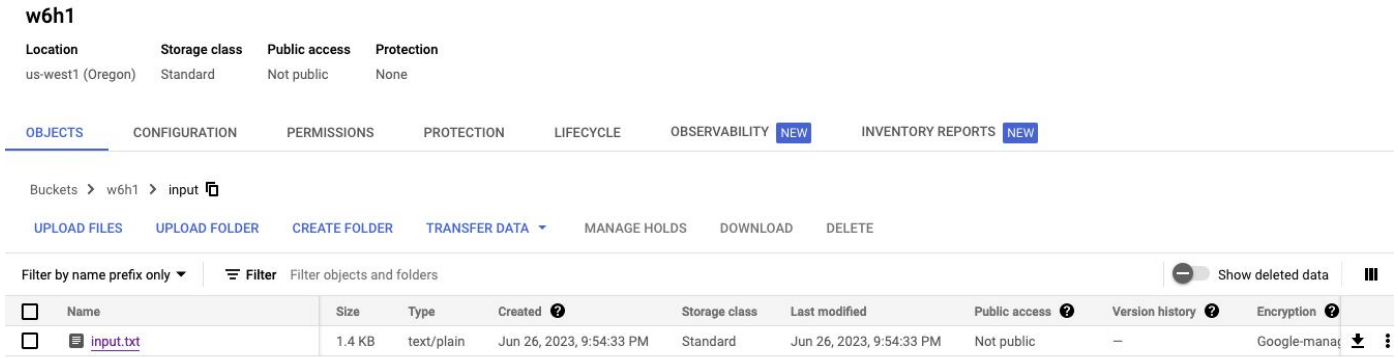
CS570 > input.py > ...

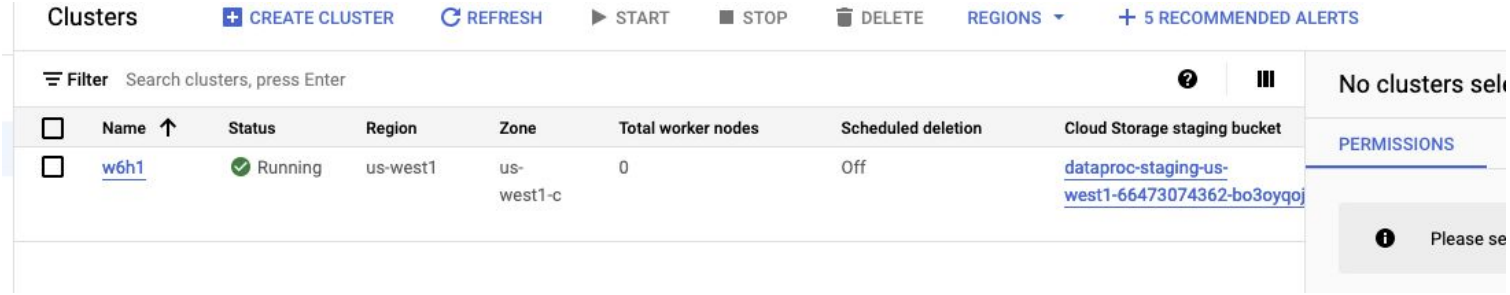
```python
import random

# Generate 200 random coordinates
coordinates = [(random.randint(1, 9), random.randint(1, 9)) for _ in range(200)]

# Write the coordinates to the input file
with open("input.txt", "w") as file:
    for x, y in coordinates:
        print(f"({x}, {y})")
        file.write(f"({x}, {y})\n")
```

input.txt

```
(4, 7)
(8, 5)
(9, 1)
(2, 7)
(8, 3)
(4, 8)
(6, 6)
(1, 9)
(9, 3)
(4, 9)
(4, 1)
(5, 9)
(6, 6)
(8, 3)
(4, 6)
(6, 9)
(7, 2)
(2, 7)
(4, 3)
(5, 6)
(9, 6)
(5, 2)
(3, 8)
(8, 5)
(1, 4)
(6, 5)
(8, 6)
(6, 1)
(1, 5)
(8, 3)
(8, 8)
(1, 9)
(8, 2)
(9, 9)
(9, 5)
(5, 7)
(5, 9)
(9, 6)
(1, 4)
(8, 2)
(7, 7)
(1, 7)
(8, 5)
(8, 5)
(2, 9)
(9, 1)
(4, 2)
(7, 2)
(8, 8)
(3, 2)
(2, 5)
```

Create a Bucket and dataproc Clusters on GCP





PySpark Pi calculation program

pi.py × input.py

pi.py

```python
from pyspark.sql import SparkSession
import sys

# Create a SparkSession
spark = SparkSession.builder.appName("PiEstimation").getOrCreate()

if len(sys.argv) != 3:
    raise Exception("Exactly 2 arguments are required: <inputUri> <outputUri>")
inputUri = sys.argv[1]
outputUri = sys.argv[2]

# Read the input file containing 20 coordinates (x, y)
coordinates = spark.read.text(inputUri)

# Define the function to calculate if a point is inside the circle
# Radius = 5
def points(row):
    x, y = map(float, row.value[1:-1].split(','))
    if x**2 + y**2 <= 5**2:
        return "inside"
    else:
        return "outside"

# Calculate the number of points inside and outside the unit circle
point_counts = coordinates.rdd.map(points).countByValue()

# Get the count of points inside the circle
inside_circle_count = point_counts.get("inside", 0)

# Get the count of points outside the circle
outside_circle_count = point_counts.get("outside", 0)

# Calculate the total number of points
total_count = coordinates.count()

# Estimate the value of pi
pi_estimate = 4.0 * inside_circle_count / total_count

# Print
print("Points inside the circle:", inside_circle_count)
print("Points outside the circle:", outside_circle_count)
print("Pi is approximately:", pi_estimate)

# Stop the SparkSession
spark.stop()
```

Execution

gcloud dataproc jobs submit pyspark pi.py --cluster=w6h1 --region=us-west1 -- gs://w6h1/input/input.txt gs://w6h1/output

```
jfang757@cloudshell:~ (cs570jf)$ gcloud dataproc jobs submit pyspark pi.py --cluster=w6h1 --region=us-west1 -- gs://w6h1/input/input.txt gs://w6h1/output
Job [c408d371e38740a7a7e7a98fcb985812] submitted.
Waiting for job output...
23/06/27 04:54:54 INFO org.apache.spark.SparkEnv: Registering MapOutputTracker
23/06/27 04:54:54 INFO org.apache.spark.SparkEnv: Registering BlockManagerMaster
23/06/27 04:54:55 INFO org.apache.spark.SparkEnv: Registering BlockManagerMasterHeartbeat
23/06/27 04:54:55 INFO org.apache.spark.SparkEnv: Registering OutputCommitCoordinator
23/06/27 04:54:55 INFO org.sparkproject.jetty.util.log: Logging initialized @2825ms to org.sparkproject.jetty.util.log.Slf4jLog
23/06/27 04:54:55 INFO org.sparkproject.jetty.server.Server: jetty-9.4.40.v20210413; built: 2021-04-13T20:42:42.668Z; git: b881a572662e1943a14ae12e7e1207989f218b74; jvm 1.8.0_372-b07
23/06/27 04:54:55 INFO org.sparkproject.jetty.server.Server: Started @2919ms
23/06/27 04:54:55 INFO org.sparkproject.jetty.server.AbstractConnector: Started ServerConnector@2f588b80{HTTP/1.1, (http/1.1)}{0.0.0.0:40317}
23/06/27 04:54:55 INFO org.apache.hadoop.yarn.client.RMProxy: Connecting to ResourceManager at w6h1-m/10.138.0.9:8032
23/06/27 04:54:56 INFO org.apache.hadoop.yarn.client.AHSProxy: Connecting to Application History server at w6h1-m/10.138.0.9:10200
23/06/27 04:54:57 INFO org.apache.hadoop.conf.Configuration: resource-types.xml not found
23/06/27 04:54:57 INFO org.apache.hadoop.yarn.util.resource.ResourceUtils: Unable to find 'resource-types.xml'.
23/06/27 04:54:57 INFO org.apache.hadoop.yarn.client.api.impl.YarnClientImpl: Submitted application application_1687838987543_0007
23/06/27 04:54:58 INFO org.apache.hadoop.yarn.client.RMProxy: Connecting to ResourceManager at w6h1-m/10.138.0.9:8030
23/06/27 04:55:01 INFO com.google.cloud.hadoop.repackaged.gcs.com.google.cloud.hadoop.gcsio.GoogleCloudStorageImpl: Ignoring exception of type GoogleJsonResponseException; verified object already exists
 with desired state.
Points inside the circle: 37
Points outside the circle: 163
Pi is approximately: 0.74
23/06/27 04:55:12 INFO org.sparkproject.jetty.server.AbstractConnector: Stopped Spark@2f588b80{HTTP/1.1, (http/1.1)}{0.0.0.0:0}
Job [c408d371e38740a7a7e7a98fcb985812] finished successfully.
done: true
driverControlFilesUri: gs://dataproc-staging-us-west1-66473074362-bo3oyqoj/google-cloud-dataproc-metainfo/3bb07439-774d-4fc1-a754-c1d2bf261087/jobs/c408d371e38740a7a7e7a98fcb985812/
driverOutputResourceUri: gs://dataproc-staging-us-west1-66473074362-bo3oyqoj/google-cloud-dataproc-metainfo/3bb07439-774d-4fc1-a754-c1d2bf261087/jobs/c408d371e38740a7a7e7a98fcb985812/driveroutput
jobUuid: a12bec22-ff86-34ef-84ea-75db6a270642
placement:
  clusterName: w6h1
  clusterUuid: 3bb07439-774d-4fc1-a754-c1d2bf261087
pysparkJob:
  args:
  - gs://w6h1/input/input.txt
  - gs://w6h1/output
  mainPythonFileUri: gs://dataproc-staging-us-west1-66473074362-bo3oyqoj/google-cloud-dataproc-metainfo/3bb07439-774d-4fc1-a754-c1d2bf261087/jobs/c408d371e38740a7a7e7a98fcb985812/staging/pi.py
reference:
  jobId: c408d371e38740a7a7e7a98fcb985812
  projectId: cs570jf
status:
  state: DONE
  stateStartTime: '2023-06-27T04:55:16.662221Z'
statusHistory:
- state: PENDING
  stateStartTime: '2023-06-27T04:54:51.461208Z'
- state: SETUP_DONE
  stateStartTime: '2023-06-27T04:54:51.490776Z'
- details: Agent reported job success
  state: RUNNING
  stateStartTime: '2023-06-27T04:54:51.673991Z'
yarnApplications:
- name: PiEstimation
  progress: 1.0
  state: FINISHED
  trackingUrl: http://w6h1-m:8088/proxy/application_1687838987543_0007/
jfang757@cloudshell:~ (cs570jf)$
```

Result

I use 200 pairs with a radius of 5 for this project, and the results is

Inside: 37

Outside: 163

Pi: 0.74

```
Points inside the circle: 37
Points outside the circle: 163
Pi is approximately: 0.74
```

Conclusion

The result 0.74 is far off Pi, but I only use 200 pairs of numbers. If we increase the number of pairs to 2000 or more, the result will be much closer to the Pi.

References

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

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