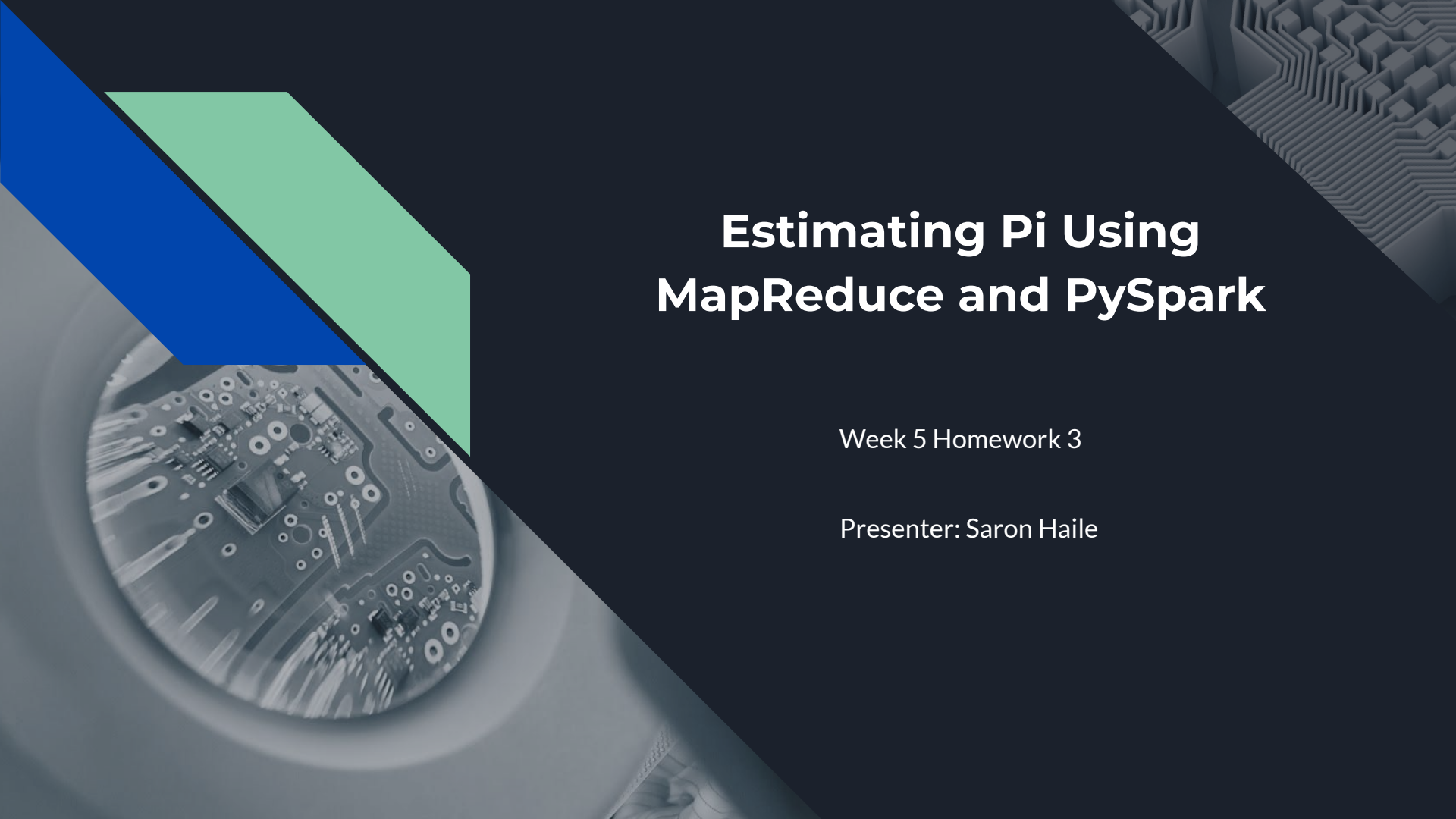


Estimating Pi Using MapReduce and PySpark

Week 5 Homework 3

Presenter: Saron Haile

Table of Content

Introduction

- The objective of this project is to calculate the value of Pi using two different approaches: MapReduce and PySpark.

- The project involves setting up a VM instance on Google Cloud Platform (GCP), writing Java programs to generate random dots, implementing MapReduce to calculate Pi, and enhancing the result using PySpark.

Design

1. **Step 1**: MapReduce approach to calculate Pi:
    - Generate random dot pairs using a Java program.
    - Implement MapReduce to calculate Pi based on the generated dots.
    - Calculate Pi using the MapReduce result.
    -
2. **Step 2**: Enhance the result using PySpark:
    - Implement Pi Calculation using PySpark on GCP.
    - Compare the results from MapReduce and PySpark.

Job: Pi

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							

Implementation-MapReduce

Methodology: Implemented the Pi estimation algorithm using Java and Hadoop MapReduce.

Architecture: Developed two main components - random number generation and Pi calculation - within the MapReduce framework.

Code Structure: Divided the code into mapper and reducer tasks, each handling specific parts of the computation.

Data Flow: Illustrated the flow of data through the MapReduce pipeline during Pi estimation.

Implementation-MapReduce

1. Start VM Instance on GCP

```
shaile32266@cs570webserver:~$ ssh localhost
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.15.0-1060-gcp x86_64)

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

 System information as of Thu Jun  6 04:08:28 UTC 2024

  System load:  0.25               Processes:             111
  Usage of /:   57.9% of 9.51GB    Users logged in:       0
  Memory usage: 20%                IPv4 address for ens4: 10.212.0.2
  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.

11 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

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


Last login: Thu Jun  6 04:08:29 2024 from 35.235.241.130
```

Implementation-MapReduce

Create Directory: PiProject

```
shaile32266@cs570webserver:~$ ls
PiProject  hadoop-3.4.0  hadoop-3.4.0.tar.gz
shaile32266@cs570webserver:~$
```

```
shaile32266@cs570webserver:~/hadoop-3.4.0$ cd ../PiProject
shaile32266@cs570webserver:~/PiProject$ mkdir input
shaile32266@cs570webserver:~/PiProject$ ls
CalculatePi.java  CalculatePiMR.java  GenerateDots.java  input
```

Implementation-MapReduce

Create GenerateDots.java

```
shaile32266@cs570webserver:~/PiProject$ nano GenerateDots.java
```

```
  GNU nano 4.8                                GenerateDots.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));
        }
    }
}
```

Implementation-MapReduce

Create CalculatePiMR.java

GNU nano 4.8 CalculatePiMR.java Modifi

```
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");
```

GNU nano 4.8 CalculatePiMR.java Modified

```
                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);
   }
}
```

Implementation-MapReduce

Create CalculatePi.java

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

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

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

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

                bufferedReader.close();
        }
}
```

1. Format the file system

```
shaile32266@cs570webserver:~/hadoop-3.4.0$ bin/hdfs namenode -format
2024-06-06 04:26:34,874 INFO namenode.NameNode: STARTUP_MSG:
/************************************************************
STARTUP_MSG: Starting NameNode
STARTUP_MSG:   host = cs570webserver.me-central1-a.c.cs570-project1-425210.internal/10.212.0.2
STARTUP_MSG:   args = [-format]
STARTUP_MSG:   version = 3.4.0
STARTUP_MSG:   classpath = /home/shaile32266/hadoop-3.4.0/etc/hadoop:/home/shaile32266/hadoop-3.4.0/share/hadoo
p/common/lib/kerb-client-2.0.3.jar:/home/shaile32266/hadoop-3.4.0/share/hadoop/common/lib/curator-client-5.2.0.
jar:/home/shaile32266/hadoop-3.4.0/share/hadoop/common/lib/netty-resolver-dns-native-macos-4.1.100.Final-osx-aa
rch_64.jar:/home/shaile32266/hadoop-3.4.0/share/hadoop/common/lib/netty-codec-http-4.1.100.Final.jar:/home/shai
le32266/hadoop-3.4.0/share/hadoop/common/lib/netty-codec-4.1.100.Final.jar:/home/shaile32266/hadoop-3.4.0/share
/hadoop/common/lib/kerb-util-2.0.3.jar:/home/shaile32266/hadoop-3.4.0/share/hadoop/common/lib/jetty-security-9.
4.53.v20231009.jar:/home/shaile32266/hadoop-3.4.0/share/hadoop/common/lib/jakarta.activation-api-1.2.1.jar:/hom
e/shaile32266/hadoop-3.4.0/share/hadoop/common/lib/netty-codec-stomp-4.1.100.Final.jar:/home/shaile32266/hadoop
-3.4.0/share/hadoop/common/lib/jetty-server-9.4.53.v20231009.jar:/home/shaile32266/hadoop-3.4.0/share/hadoop/co
mmon/lib/kerby-xdr-2.0.3.jar:/home/shaile32266/hadoop-3.4.0/share/hadoop/common/lib/netty-transport-native-unix
-common-4.1.100.Final.jar:/home/shaile32266/hadoop-3.4.0/share/hadoop/common/lib/netty-transport-classes-epoll-
4.1.100.Final.jar:/home/shaile32266/hadoop-3.4.0/share/hadoop/common/lib/netty-handler-proxy-4.1.100.Final.jar:
/home/shaile32266/hadoop-3.4.0/share/hadoop/common/lib/zookeeper-jute-3.8.3.jar:/home/shaile32266/hadoop-3.4.0/
share/hadoop/common/lib/commons-text-1.10.0.jar:/home/shaile32266/hadoop-3.4.0/share/hadoop/common/lib/jaxb-api
-2.2.11.jar:/home/shaile32266/hadoop-3.4.0/share/hadoop/common/lib/netty-handler-ssl-ocsp-4.1.100.Final.jar:/ho
me/shaile32266/hadoop-3.4.0/share/hadoop/common/lib/netty-codec-socks-4.1.100.Final.jar:/home/shaile32266/hadoo
p-3.4.0/share/hadoop/common/lib/jetty-http-9.4.53.v20231009.jar:/home/shaile32266/hadoop-3.4.0/share/hadoop/com
mon/lib/netty-resolver-4.1.100.Final.jar:/home/shaile32266/hadoop-3.4.0/share/hadoop/common/lib/hadoop-shaded-g
uava-1.2.0.jar:/home/shaile32266/hadoop-3.4.0/share/hadoop/common/lib/netty-transport-native-epoll-4.1.100.Fina
l.jar:/home/shaile32266/hadoop-3.4.0/share/hadoop/common/lib/jetty-util-9.4.53.v20231009.jar:/home/shaile32266/
hadoop-3.4.0/share/hadoop/common/lib/jackson-annotations-2.12.7.jar:/home/shaile32266/hadoop-3.4.0/share/hadoop
/common/lib/nimbus-jose-jwt-9.31.jar:/home/shaile32266/hadoop-3.4.0/share/hadoop/common/lib/kerby-pkix-2.0.3.ja
r:/home/shaile32266/hadoop-3.4.0/share/hadoop/common/lib/gson-2.9.0.jar:/home/shaile32266/hadoop-3.4.0/share/ha
doop/common/lib/commons-lang3-3.12.0.jar:/home/shaile32266/hadoop-3.4.0/share/hadoop/common/lib/slf4j-api-1.7.3
6.jar:/home/shaile32266/hadoop-3.4.0/share/hadoop/common/lib/commons-io-2.14.0.jar:/home/shaile32266/hadoop-3.4
.0/share/hadoop/common/lib/jetty-util-ajax-9.4.53.v20231009.jar:/home/shaile32266/hadoop-3.4.0/share/hadoop/com
mon/lib/commons-compress-1.24.0.jar:/home/shaile32266/hadoop-3.4.0/share/hadoop/common/lib/animal-sniffer-annot
```

2. Start daemons

```
shaile32266@cs570webserver:~/hadoop-3.4.0$ sbin/start-dfs.sh
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [cs570webserver]
```

3. Test Connection to localhost

```
shaile32266@cs570webserver:~/hadoop-3.4.0$ wget http://localhost:9870/
--2024-06-06 03:33:46--  http://localhost:9870/
Resolving localhost (localhost)... 127.0.0.1
Connecting to localhost (localhost)|127.0.0.1|:9870... connected.
HTTP request sent, awaiting response... 302 Found
Location: http://localhost:9870/index.html [following]
--2024-06-06 03:33:46--  http://localhost:9870/index.html
Reusing existing connection to localhost:9870.
HTTP request sent, awaiting response... 200 OK
Length: 1079 (1.1K) [text/html]
Saving to: 'index.html.2'

index.html.2                100%[=========================================>]   1.05K  --.-KB/s    in 0s

2024-06-06 03:33:46 (149 MB/s) - 'index.html.2' saved [1079/1079]

shaile32266@cs570webserver:~/hadoop-3.4.0$
```

4. Compile and run GenerateDots.java

```
shaile32266@cs570webserver:~/PiProject$ java GenerateDots 5 1000 > ./input/dots.txt
shaile32266@cs570webserver:~/PiProject$
```

```
shaile32266@cs570webserver:~/PiProject$ cat ./input/dots.txt
6.890236293933495 2.29262825482889 5.0
9.80124284804244 3.3172110357801543 5.0
0.6849641932681094 4.570355558484421 5.0
5.803435467918623 9.944334325006977 5.0
7.2119978959019 9.25736558652102 5.0
1.9538325574863558 7.762783634803175 5.0
1.7920460920690429 0.4164656285885615 5.0
3.382098749900754 6.8494865847647235 5.0
2.5659449542439736 2.8875756071136016 5.0
5.578022813585738 9.337065289691974 5.0
0.14415659622645793 7.57374751395122 5.0
6.010720436165089 3.9626906438251397 5.0
4.677501168258078 4.3212290918206255 5.0
3.869297899374283 4.312815344813795 5.0
2.445380033911823 8.906322758428326 5.0
4.420644234727814 1.7150184525119105 5.0
6.997192186306674 4.284379655432284 5.0
```

5. Copy file from local to hadoop

```
shaile32266@cs570webserver:~$ cd hadoop-3.4.0
shaile32266@cs570webserver:~/hadoop-3.4.0$ bin/hdfs dfs -mkdir /user
shaile32266@cs570webserver:~/hadoop-3.4.0$ bin/hdfs dfs -mkdir /user/shaile32266
shaile32266@cs570webserver:~/hadoop-3.4.0$ bin/hdfs dfs -mkdir /user/shaile32266/PiProject
shaile32266@cs570webserver:~/hadoop-3.4.0$ bin/hdfs dfs -mkdir /user/shaile32266/PiProject/Input
shaile32266@cs570webserver:~/hadoop-3.4.0$ bin/hdfs dfs -put ../PiProject/Input/* PiProject/Input
shaile32266@cs570webserver:~/hadoop-3.4.0$ bin/hdfs dfs -ls PiProject/Input
Found 1 items
-rw-r--r--   1 shaile32266 supergroup      40554 2024-06-06 01:55 PiProject/Input/dots.txt
shaile32266@cs570webserver:~/hadoop-3.4.0$
```

6. Compile MapReduce program in Hadoop

```
shaile32266@cs570webserver:~/hadoop-3.4.0$ bin/hadoop jar ~/hadoop-3.4.0/share/hadoop/mapreduce/hadoop-
mapreduce-client-core-3.4.0.jar com.sun.tools.javac.Main ~/PiProject/CalculatePiMR.java
Note: /home/shaile32266/PiProject/CalculatePiMR.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
shaile32266@cs570webserver:~/hadoop-3.4.0$
```

7. Run MapReduce Program

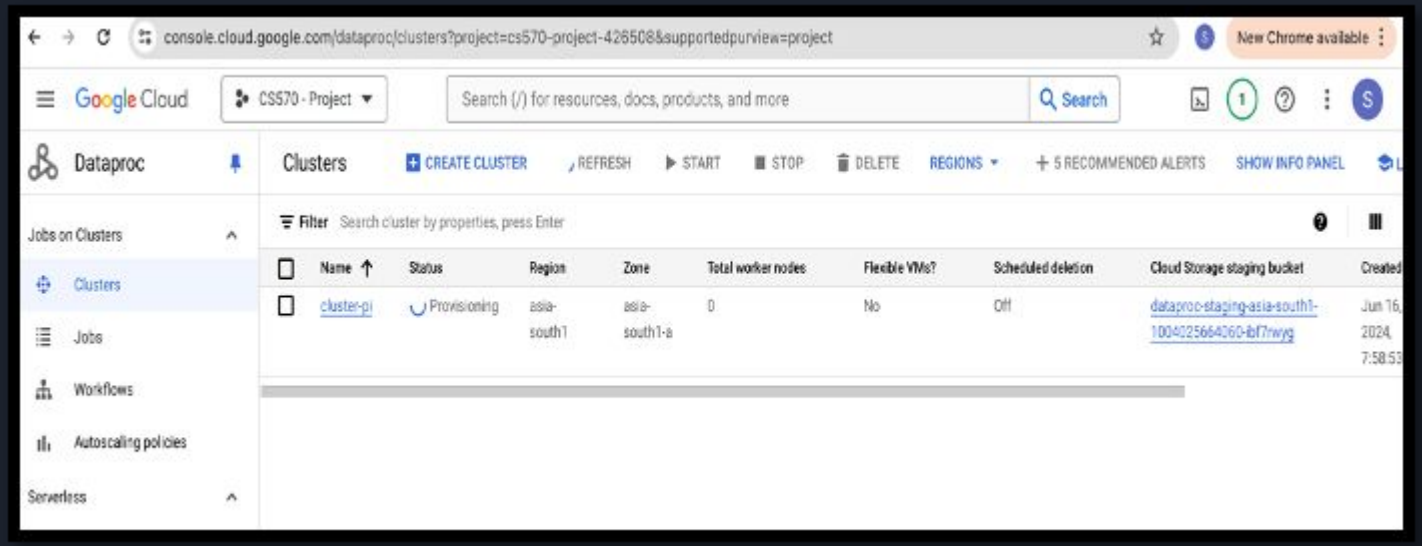
```
shaile32266@cs570webserver:~/hadoop-3.4.0$ bin/hadoop jar ~/PiProject/pi.jar CalculatePiMR /user/shaile
32266/PiProject/Input /user/shaile32266/PiProject/Output
2024-06-06 02:39:32,413 INFO impl.MetricsConfig: Loaded properties from hadoop-metrics2.properties
2024-06-06 02:39:32,576 INFO impl.MetricsSystemImpl: Scheduled Metric snapshot period at 10 second(s).
2024-06-06 02:39:32,576 INFO impl.MetricsSystemImpl: JobTracker metrics system started
2024-06-06 02:39:32,841 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not perf
ormed. Implement the Tool interface and execute your application with ToolRunner to remedy this.
2024-06-06 02:39:33,112 INFO input.FileInputFormat: Total input files to process : 1
2024-06-06 02:39:33,146 INFO mapreduce.JobSubmitter: number of splits:1
2024-06-06 02:39:33,420 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_local2127946298_000
1
2024-06-06 02:39:33,421 INFO mapreduce.JobSubmitter: Executing with tokens: []
2024-06-06 02:39:33,672 INFO mapreduce.Job: The url to track the job: http://localhost:8080/
2024-06-06 02:39:33,674 INFO mapreduce.Job: Running job: job_local2127946298_0001
2024-06-06 02:39:33,683 INFO mapred.LocalJobRunner: OutputCommitter set in config null
2024-06-06 02:39:33,705 INFO output.PathOutputCommitterFactory: No output committer factory defined, de
faulting to FileOutputCommitterFactory
2024-06-06 02:39:33,707 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 2
2024-06-06 02:39:33,707 INFO output.FileOutputCommitter: FileOutputCommitter skip cleanup _temporary fo
lders under output directory:false, ignore cleanup failures: false
2024-06-06 02:39:33,708 INFO mapred.LocalJobRunner: OutputCommitter is org.apache.hadoop.mapreduce.lib.
output.FileOutputCommitter
2024-06-06 02:39:33,808 INFO mapred.LocalJobRunner: Waiting for map tasks
2024-06-06 02:39:33,809 INFO mapred.LocalJobRunner: Starting task: attempt_local2127946298_0001_m_00000
0_0
2024-06-06 02:39:33,847 INFO output.PathOutputCommitterFactory: No output committer factory defined, de
faulting to FileOutputCommitterFactory
2024-06-06 02:39:33,847 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 2
2024-06-06 02:39:33,847 INFO output.FileOutputCommitter: FileOutputCommitter skip cleanup _temporary fo
lders under output directory:false, ignore cleanup failures: false
2024-06-06 02:39:33,877 INFO mapred.Task:  Using ResourceCalculatorProcessTree : [ ]
2024-06-06 02:39:33,885 INFO mapred.MapTask: Processing split: hdfs://localhost:9000/user/shaile32266/P
iProject/Input/dots.txt:0+40554
2024-06-06 02:39:34,092 INFO mapred.MapTask: (EQUATOR) 0 kvi 26214396(104857584)
2024-06-06 02:39:34,092 INFO mapred.MapTask: mapreduce.task.io.sort.mb: 100
2024-06-06 02:39:34,092 INFO mapred.MapTask: soft limit at 83886080
2024-06-06 02:39:34,092 INFO mapred.MapTask: bufstart = 0; bufvoid = 104857600
2024-06-06 02:39:34,092 INFO mapred.MapTask: kvstart = 26214396; length = 6553600
2024-06-06 02:39:34,099 INFO mapred.MapTask: Map output collector class = org.apache.hadoop.mapred.MapT
ask$MapOutputBuffer
```

Implementation-Pyspark

1. **Create Bucket**

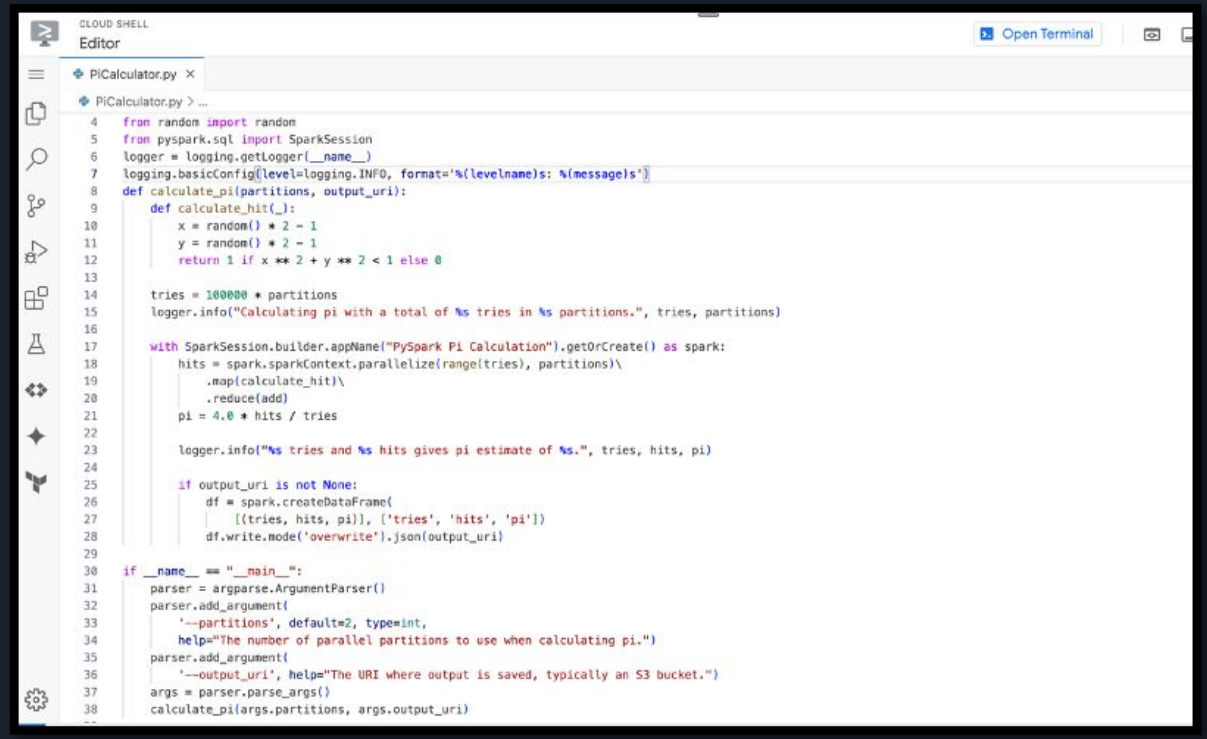
Implementation-Pyspark

2. Create Cluster



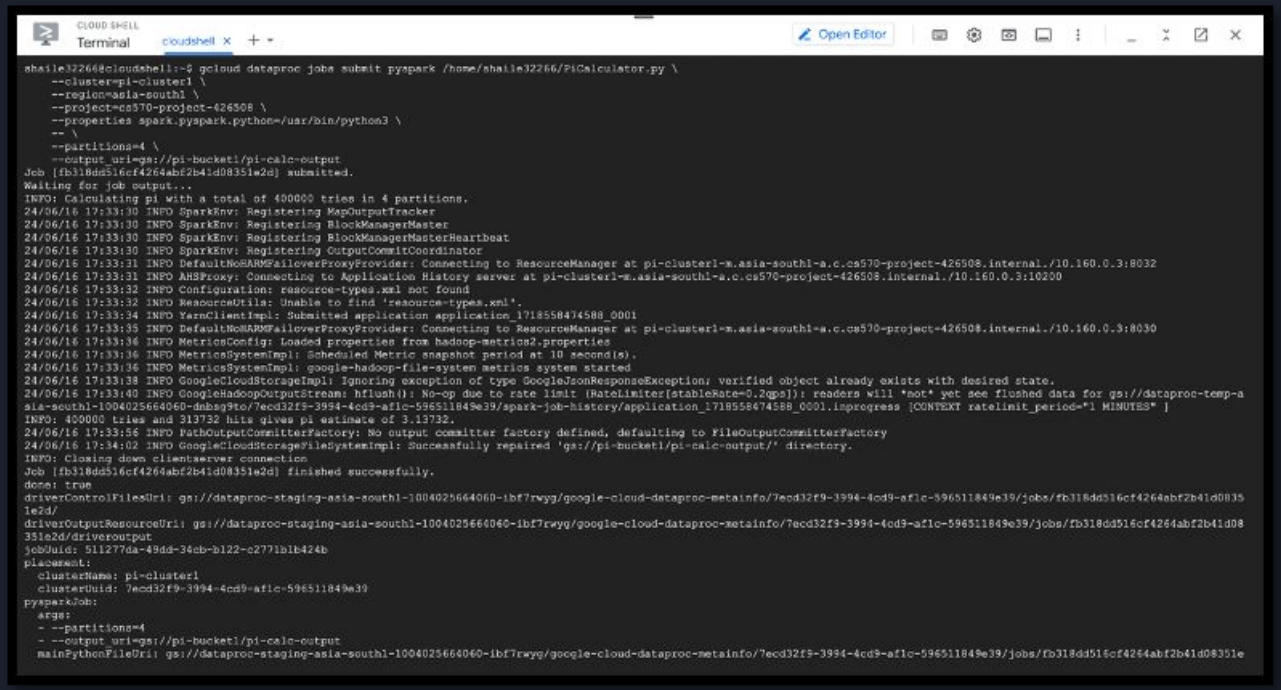
Implementation-Pyspark

3. Create Python code to calculate Pi



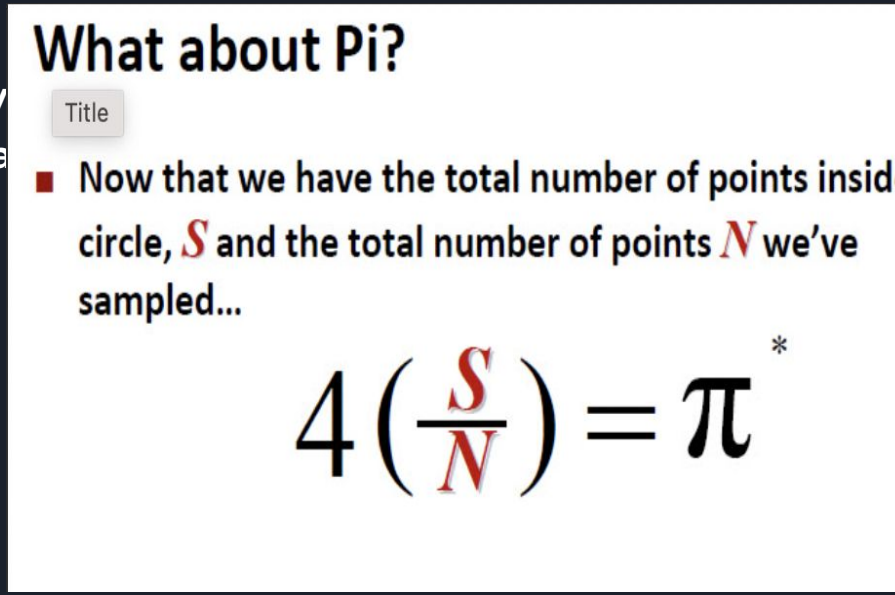
Implementation-Pyspark

4. Submit the job under cloud shell terminal



Testing

- **Integration Testing**: Tested the integration of the random number generator and Pi calculation components within the MapReduce framework.
- **As expected, the PI value calculated by Py**
  of pi which is 3.13732 as compared to the va
  which was 3.104.
-



MapReduce Result

View the output

```
shaile32266@ubuntuwebserver:~/hadoop-3.3.6$ cd ../PiProject
shaile32266@ubuntuwebserver:~/PiProject$ java CalculatePi Output
Inside  776
Outside 224
PI value is: 3.104
shaile32266@ubuntuwebserver:~/PiProject$
```

Pyspark Result

View the output folder

```
shaile32266@cloudshell:~$ gsutil ls gs://pi-bucket1/pi-calc-output/
gs://pi-bucket1/pi-calc-output/
gs://pi-bucket1/pi-calc-output/_SUCCESS
gs://pi-bucket1/pi-calc-output/part-00000-a19da3b4-a308-4f13-b912-5967b453ad79-c000.json
gs://pi-bucket1/pi-calc-output/part-00003-a19da3b4-a308-4f13-b912-5967b453ad79-c000.json
```

```
shaile32266@cloudshell:~$ gsutil cat gs://pi-bucket1/pi-calc-output/part-00000-a19da3b4-a308-4f13-b912-5967b453ad79-c000.json
shaile32266@cloudshell:~$ gsutil cat gs://pi-bucket1/pi-calc-output/part-00003-a19da3b4-a308-4f13-b912-5967b453ad79-c000.json | jq
{
  "tries": 400000,
  "hits": 313732,
  "pi": 3.13732
}
```

Pi Accuracy

As expected, the PI value calculated by PySpark is closer to the actual value of pi which is 3.13732 as compared to the value of pi calculated by MapReduce which was 3.104.

Enhancement

- **Optimization**: Explore optimization techniques to improve the efficiency of Pi estimation, such as parallelization and algorithmic optimizations.
- **Visualization**: Develop visualizations to represent the distribution of dart throws and the estimation of Pi for better understanding.
- **Error Handling**: Enhance error handling mechanisms to handle edge cases and exceptions gracefully.

Conclusion

- Successfully implemented Pi estimation using MapReduce and PySpark.
- Overcame challenges related to data distribution, synchronization, and fault tolerance

Link to Github

https://github.com/Sharon20222/Cloud-Computing/tree/main/PySpark/Pi

References

- https://hadoop.apache.org/docs/stable/hadoop-project-dist/hadoop-common/SingleCluster.html

Thank you!