Assignment3 – Report

Part-1:

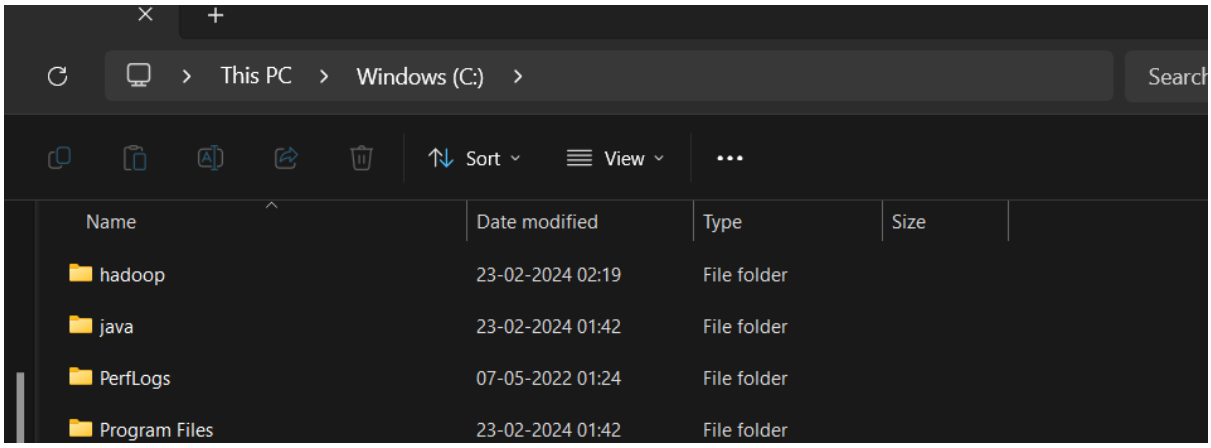
This part contains Hadoop setup steps and running hadoop's example package to calculate the value of pi.

- First check if oracle java-8 compatible version is properly installed on your system. If not, need to download and install:

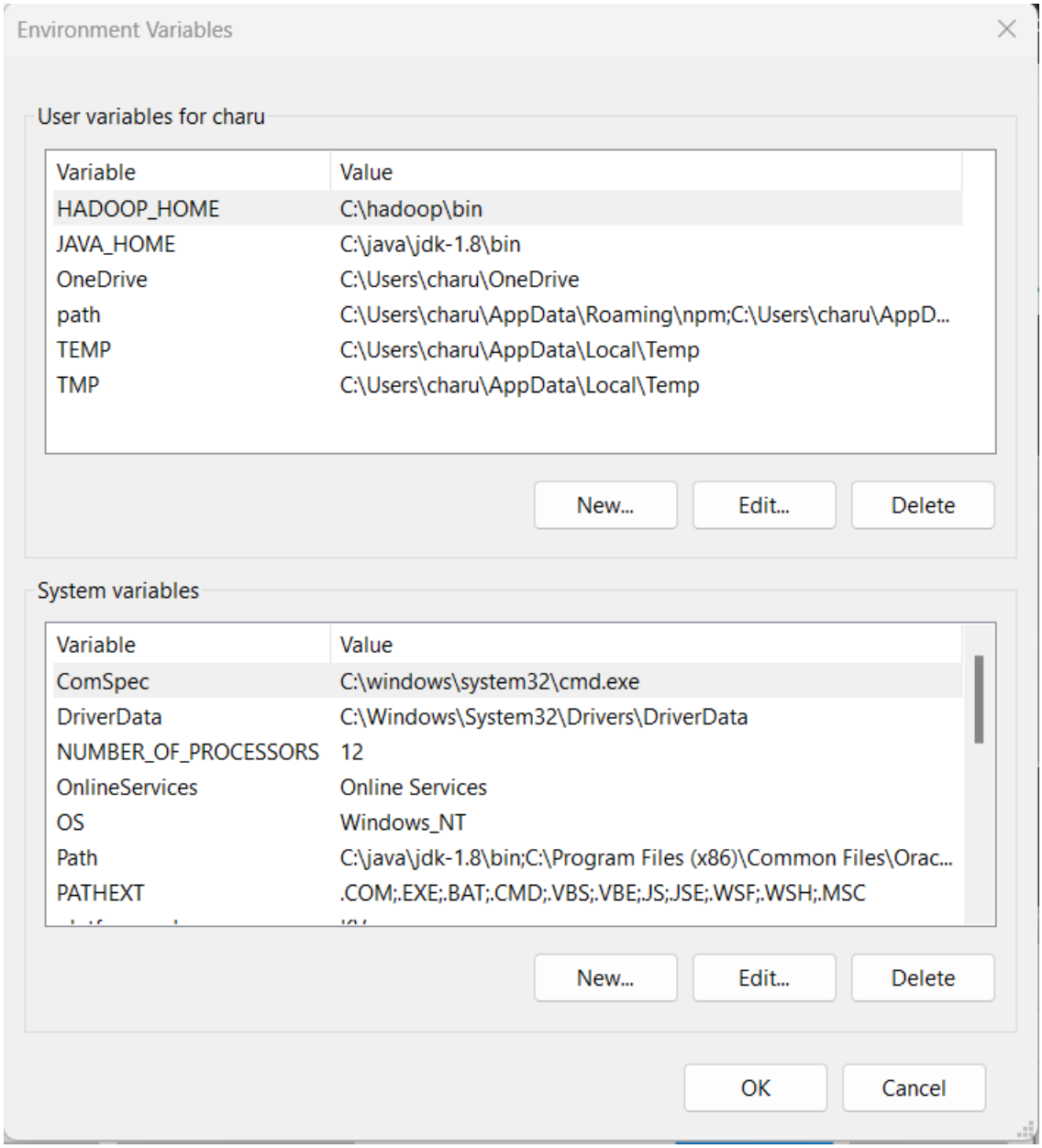
```
C:\Users\charu>javac -version
javac 1.8.0_401

C:\Users\charu>java -version
java version "1.8.0_401"
Java(TM) SE Runtime Environment (build 1.8.0_401-b10)
Java HotSpot(TM) 64-Bit Server VM (build 25.401-b10, mixed mode)
```

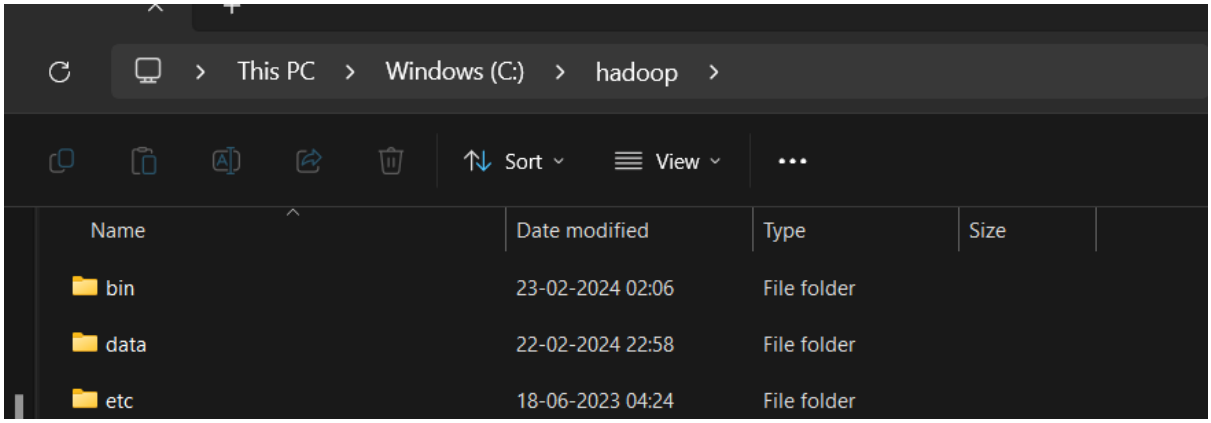
- Next download Hadoop 3.3.6 bin from the official site and extract the package in C:\hadoop:

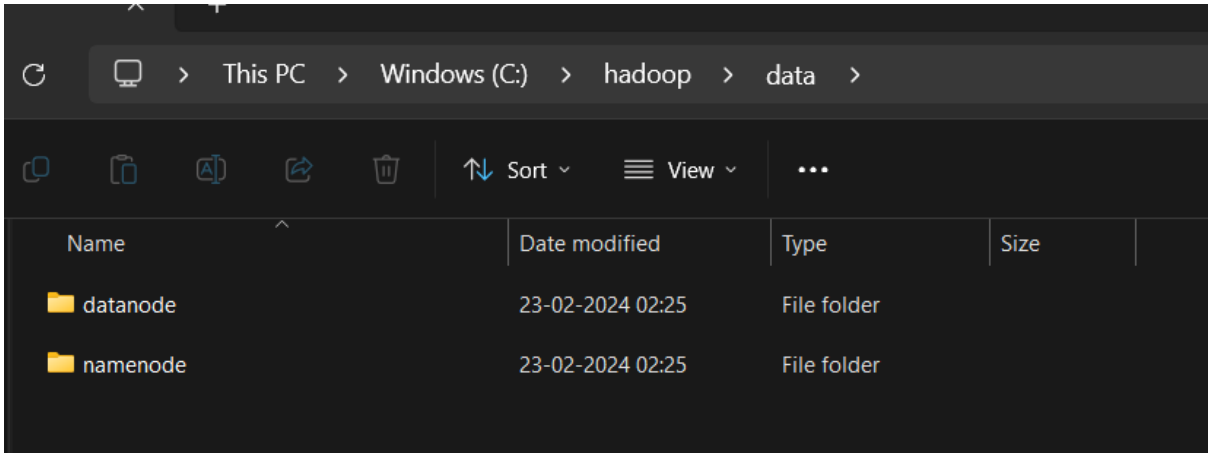


- Now check and add environment variables ‘JAVA_HOME’, ‘HADOOP_HOME’ and the path of bin folders for both from advanced system settings as shown below:



- Now create a data folder inside Hadoop folder and two folders, namenode and datanode inside the data folders:





- Edit the below files to add configuration and JAVA_HOME path as shown:
  1. Core-site.xml

```
<configuration>
   <property>
       <name>fs.defaultFS</name>
       <value>hdfs://localhost:9000</value>
   </property>
</configuration>
```

  2. Mapred-site.xml

```
<configuration>
   <property>
       <name>mapreduce.framework.name</name>
       <value>yarn</value>
   </property>
</configuration>
```

3. Hdfs-site.xml

```
<configuration>
   <property>
       <name>dfs.replication</name>
       <value>1</value>
   </property>
   <property>
       <name>dfs.namenode.name.dir</name>
       <value>C:\hadoop\data\namenode</value>
   </property>
   <property>
       <name>dfs.datanode.data.dir</name>
       <value>C:\hadoop\data\datanode</value>
   </property>
</configuration>
```

4. Yarn-site.xml

```
<configuration>

<!-- Site specific YARN configuration properties -->
   <property>
        <name>yarn.nodemanager.aux-services</name>
        <value>mapreduce_shuffle</value>
   </property>
   <property>
          <name>yarn.nodemanager.auxservices.mapreduce.shuffle.class</name>
    <value>org.apache.hadoop.mapred.ShuffleHandler</value>
   </property>

</configuration>
```

5. Hadoop-env.cmd

```
@rem The java implementation to use.  Required.
set JAVA_HOME=C:\java\jdk-1.8
```

- Now open cmd and format namenode folder using command: hdfs namenode –format

```
C:\Windows\System32>hdfs namenode -format
2024-02-23 02:19:24,556 INFO namenode.NameNode: STARTUP_MSG:
/************************************************************
STARTUP_MSG: Starting NameNode
STARTUP_MSG:   host = Charul/192.168.1.74
STARTUP_MSG:   args = [-format]
STARTUP_MSG:   version = 3.3.6
STARTUP_MSG:   classpath = C:\hadoop\etc\hadoop;C:\hadoop\share\hadoop\common;C:\hadoop\share\hado
r;C:\hadoop\share\hadoop\common\lib\avro-1.7.7.jar;C:\hadoop\share\hadoop\common\lib\checker-qual-
-cli-1.2.jar;C:\hadoop\share\hadoop\common\lib\commons-codec-1.15.jar;C:\hadoop\share\hadoop\commo
adoop\common\lib\commons-configuration2-2.8.0.jar;C:\hadoop\share\hadoop\common\lib\commons-daemon
3-3.12.0.jar;C:\hadoop\share\hadoop\common\lib\commons-logging-1.1.3.jar;C:\hadoop\share\hadoop\co
ommon\lib\commons-text-1.10.0.jar;C:\hadoop\share\hadoop\common\lib\curator-client-5.2.0.jar;C:\ha
jar;C:\hadoop\share\hadoop\common\lib\dnsjava-2.1.7.jar;C:\hadoop\share\hadoop\common\lib\failurea
ar;C:\hadoop\share\hadoop\common\lib\hadoop-annotations-3.3.6.jar;C:\hadoop\share\hadoop\common\li
mmon\lib\hadoop-shaded-protobuf_3_7-1.1.1.jar;C:\hadoop\share\hadoop\common\lib\httpclient-4.5.13.
1.jar;C:\hadoop\share\hadoop\common\lib\jackson-annotations-2.12.7.jar;C:\hadoop\share\hadoop\comm
op\common\lib\jackson-databind-2.12.7.1.jar;C:\hadoop\share\hadoop\common\lib\jackson-mapper-asl-1
vax.servlet-api-3.1.0.jar;C:\hadoop\share\hadoop\common\lib\jaxb-api-2.2.11.jar;C:\hadoop\share\ha
e\hadoop\common\lib\jersey-core-1.19.4.jar;C:\hadoop\share\hadoop\common\lib\jersey-json-1.20.jar;
```

```
2024-02-23 02:19:27,158 INFO util.GSet: Computing capacity for map NameNodeRetryCache
2024-02-23 02:19:27,158 INFO util.GSet: VM type       = 64-bit
2024-02-23 02:19:27,158 INFO util.GSet: 0.029999999329447746% max memory 889 MB = 273.1 KB
2024-02-23 02:19:27,158 INFO util.GSet: capacity      = 2^15 = 32768 entries
2024-02-23 02:19:27,189 INFO namenode.FSImage: Allocated new BlockPoolId: BP-1417961759-192.168.1.74-1708672767183
2024-02-23 02:19:27,214 INFO common.Storage: Storage directory C:\hadoop\data\namenode has been successfully formatted.
2024-02-23 02:19:27,259 INFO namenode.FSImageFormatProtobuf: Saving image file C:\hadoop\data\namenode\current\fsimage.
2024-02-23 02:19:27,387 INFO namenode.FSImageFormatProtobuf: Image file C:\hadoop\data\namenode\current\fsimage.ckpt_00
2024-02-23 02:19:27,402 INFO namenode.NNStorageRetentionManager: Going to retain 1 images with txid >= 0
2024-02-23 02:19:27,417 INFO namenode.FSNamesystem: Stopping services started for active state
2024-02-23 02:19:27,418 INFO namenode.FSNamesystem: Stopping services started for standby state
2024-02-23 02:19:27,422 INFO namenode.FSImage: FSImageSaver clean checkpoint: txid=0 when meet shutdown.
2024-02-23 02:19:27,423 INFO namenode.NameNode: SHUTDOWN_MSG:
/************************************************************
SHUTDOWN_MSG: Shutting down NameNode at Charul/192.168.1.74
************************************************************/
```

- Navigate to sbin folder and run ‘start-all.cmd’ command to test the setup and run jps command to check:

```
C:\hadoop\sbin>jps
11536 NameNode
12176 DataNode
10684 Jps
```

- Run ‘start-yarn.cmd’ and check using jps:

```
C:\hadoop\sbin>start-yarn.cmd
starting yarn daemons

C:\hadoop\sbin>jps
11536 NameNode
12176 DataNode
22048 Jps
5284 NodeManager
18924 ResourceManager
```

- Finally run the below command to calculate value of pi:

jar C:\hadoop\share\hadoop\mapreduce\hadoop-mapreduce-examples-3.3.6.jar pi with two parameters, number of map tasks and number of samples.

```
C:\hadoop\sbin>hadoop jar C:\hadoop\share\hadoop\mapreduce\hadoop-mapreduce-examples-3.3.6.jar pi 10 1000
Number of Maps  = 10
Samples per Map = 1000
Wrote input for Map #0
Wrote input for Map #1
Wrote input for Map #2
Wrote input for Map #3
Wrote input for Map #4
Wrote input for Map #5
Wrote input for Map #6
Wrote input for Map #7
Wrote input for Map #8
Wrote input for Map #9
Starting Job
2024-02-23 02:40:52,574 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at /0.0.0.0:8032
2024-02-23 02:40:53,021 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/charu/.staging/job_1708673197994_0001
2024-02-23 02:40:53,142 INFO input.FileInputFormat: Total input files to process : 10
2024-02-23 02:40:53,181 INFO mapreduce.JobSubmitter: number of splits:10
2024-02-23 02:40:53,266 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1708673197994_0001
2024-02-23 02:40:53,266 INFO mapreduce.JobSubmitter: Executing with tokens: []
2024-02-23 02:40:53,404 INFO conf.Configuration: resource-types.xml not found
2024-02-23 02:40:53,405 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2024-02-23 02:40:53,811 INFO impl.YarnClientImpl: Submitted application application_1708673197994_0001
2024-02-23 02:40:53,877 INFO mapreduce.Job: The url to track the job: http://Charul:8088/proxy/application_1708673197994_0001/
2024-02-23 02:40:53,878 INFO mapreduce.Job: Running job: job_1708673197994_0001
2024-02-23 02:41:01,017 INFO mapreduce.Job: Job job_1708673197994_0001 running in uber mode : false
2024-02-23 02:41:01,019 INFO mapreduce.Job:  map 0% reduce 0%
```

```
                Total vcore-milliseconds taken by all map tasks=52058
                Total vcore-milliseconds taken by all reduce tasks=41
                Total megabyte-milliseconds taken by all map tasks=53
                Total megabyte-milliseconds taken by all reduce tasks
        Map-Reduce Framework
                Map input records=10
                Map output records=20
                Map output bytes=180
                Map output materialized bytes=280
                Input split bytes=1460
                Combine input records=0
                Combine output records=0
                Reduce input groups=2
                Reduce shuffle bytes=280
                Reduce input records=20
                Reduce output records=0
                Spilled Records=40
                Shuffled Maps =10
                Failed Shuffles=0
                Merged Map outputs=10
                GC time elapsed (ms)=706
                CPU time spent (ms)=2447
                Physical memory (bytes) snapshot=3804241920
                Virtual memory (bytes) snapshot=5060141056
                Total committed heap usage (bytes)=2776629248
                Peak Map Physical memory (bytes)=407756800
                Peak Map Virtual memory (bytes)=625606656
                Peak Reduce Physical memory (bytes)=239374336
                Peak Reduce Virtual memory (bytes)=413782016
        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=1180
        File Output Format Counters
                Bytes Written=97
Job Finished in 22.917 seconds
Estimated value of Pi is 3.14080000000000000000
```

Part-2:

We are running the hadoop-mapreduce-examples-3.3.6.jar pi to calculate pi using 2 parameters. First is number of maps that should be used and second is number of random points. So, in this case the value of pi is approximated by using QuasiMonteCarlo algorithm.

Code Analysis:

- In the QuasiMonteCarlo class, the estimatePi method is the driver. It takes in the number of map tasks the number of points (numPoints), a temporary directory (tmpDir), and a Hadoop Configuration object.
- In this class a job is MapReduce job is created. The mapper class QuasiMonteCarlo.QmcMapper and the reducer class is QuasiMonteCarlo.QmcReducer
- Temporary directories for input and output are set and job is configured and run.

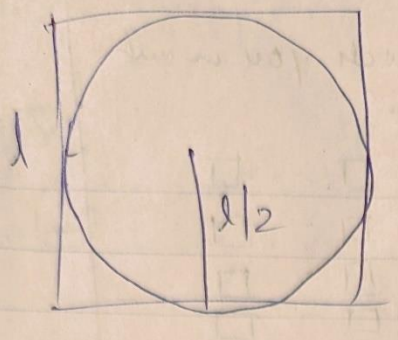
Map-part: QmcMapper

- map Method: The map method contains the logic for mapper. It takes three arguments: offset (input key), size (input value), and context.
- Halton Sequence: A QuasiMonteCarlo.HaltonSequence object is created. This used to generate random points inside the square.
- Each point (x,y) is checked to be inside or outside the circle based on its distance from the center (0.5, 0.5).
- Point is considered outside the circle if (x * x + y * y > 0.25D) otherwise inside.
- Every 1000 points, the status of the mapper is updated and results are recorded.

Reduce-Part: QmcReducer

- The reducer class aggregates the results from all mappers.
- The reduce method takes three arguments: isInside (input key), values (input values), and context.
- If sums up the total count where isInside values are true and where isInside values are false.
- The cleanup method records the aggregation which can be used to estimate pi value.

Logic used to calculate the pi value:



$$\square = l^2$$

$$\bigcirc = \pi\left(\frac{l}{2}\right)^2$$

let $K = \dfrac{\pi (l/2)^2}{l^2}$

$$K = \frac{\pi l^2}{4 l^2}$$

$$\pi = 4K$$

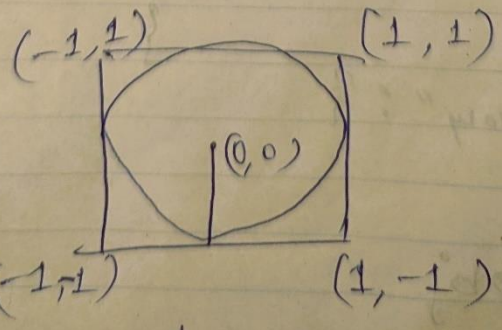
If we take randomized points in the square then,

$$K \cong \frac{\text{number of points in } \bigcirc}{\text{total number of points}}$$

then $\pi = 4K$

Since points will not cover the exact area, so we would just get an approximation of π.

Suppose



So, to generate a point we just need to generate 2 random numbers between -1 & 1