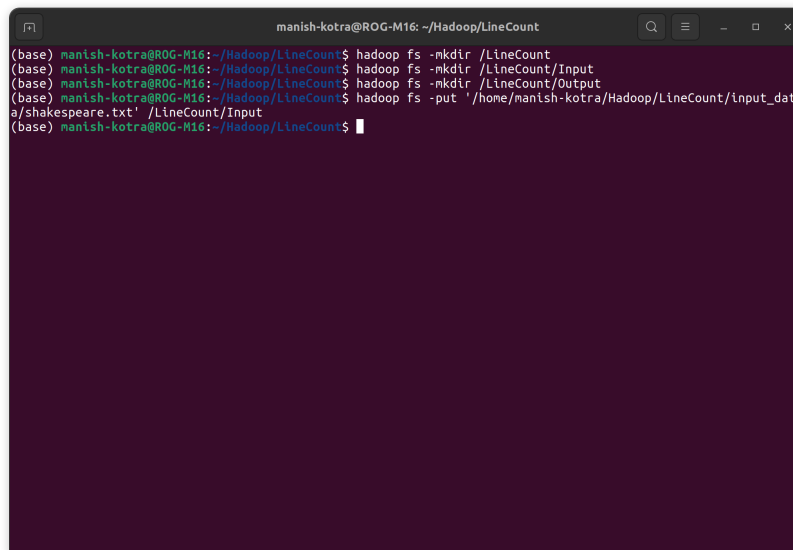


Assignment 1: Hadoop MapReduce

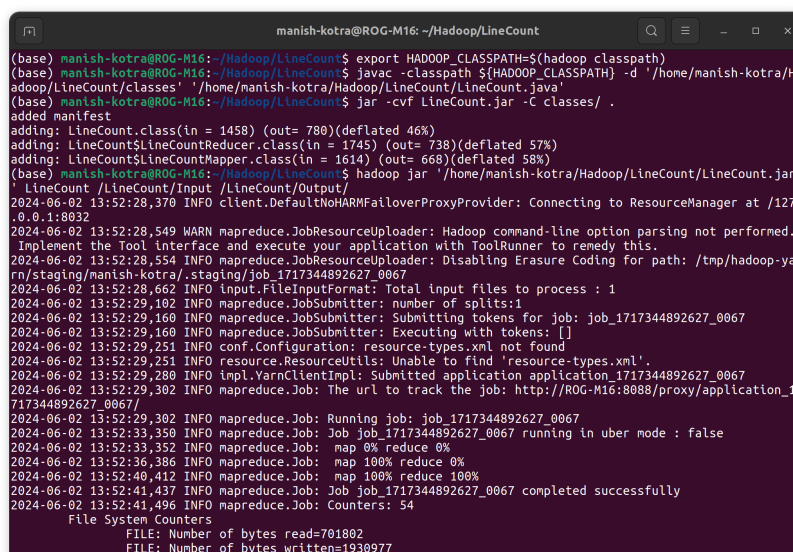
Manish Kumar - 1009645840

1. The problem setup, input commands and the outputs are given in the Figure 1-3 below. The final result for number of lines is 58483.



```
manish-kotra@ROG-M16: ~/Hadoop/LineCount
(base) manish-kotra@ROG-M16:~/Hadoop/LineCount$ hadoop fs -mkdir /LineCount
(base) manish-kotra@ROG-M16:~/Hadoop/LineCount$ hadoop fs -mkdir /LineCount/Input
(base) manish-kotra@ROG-M16:~/Hadoop/LineCount$ hadoop fs -mkdir /LineCount/Output
(base) manish-kotra@ROG-M16:~/Hadoop/LineCount$ hadoop fs -put '/home/manish-kotra/Hadoop/LineCount/input_data/shakespeare.txt' /LineCount/Input
(base) manish-kotra@ROG-M16:~/Hadoop/LineCount$
```

Figure 1. Initialize *Line Count* program



```
manish-kotra@ROG-M16: ~/Hadoop/LineCount
(base) manish-kotra@ROG-M16:~/Hadoop/LineCount$ export HADOOP_CLASSPATH=$(hadoop classpath)
(base) manish-kotra@ROG-M16:~/Hadoop/LineCount$ javac -classpath $(HADOOP_CLASSPATH) -d '/home/manish-kotra/Hadoop/LineCount/classes' '/home/manish-kotra/Hadoop/LineCount/LineCount.java'
(base) manish-kotra@ROG-M16:~/Hadoop/LineCount$ hadoop jar -cvf LineCount.jar -C classes/ .
added manifest
adding: LineCount.class(in = 1458) (out= 780)(deflated 46%)
adding: LineCount$LineCountReducer.class(in = 1745) (out= 738)(deflated 57%)
adding: LineCount$LineCountMapper.class(in = 1614) (out= 668)(deflated 58%)
manish-kotra@ROG-M16:~/Hadoop/LineCount$ hadoop jar '/home/manish-kotra/Hadoop/LineCount/LineCount.jar' /LineCount /LineCount/Input /LineCount/Output/
2024-06-02 13:52:28,370 INFO client.DefaultNoHARMAFailoverProxyProvider: Connecting to ResourceManager at /127.0.0.1:8032
2024-06-02 13:52:28,549 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.
2024-06-02 13:52:28,554 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yrn/staging/manish-kotra/.staging/job_1717344892627_0067
2024-06-02 13:52:28,662 INFO input.FileInputFormat: Total input files to process : 1
2024-06-02 13:52:29,192 INFO mapreduce.JobSubmitter: number of splits:1
2024-06-02 13:52:29,160 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1717344892627_0067
2024-06-02 13:52:29,160 INFO mapreduce.JobSubmitter: Executing with tokens: []
2024-06-02 13:52:29,251 INFO conf.Configuration: resource-types.xml not found
2024-06-02 13:52:29,251 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2024-06-02 13:52:29,280 INFO impl.YarnClientImpl: Submitted application application_1717344892627_0067
2024-06-02 13:52:29,302 INFO mapreduce.Job: The url to track the job: http://ROG-M16:8088/proxy/application_1717344892627_0067/
2024-06-02 13:52:29,302 INFO mapreduce.Job: Running job: job_1717344892627_0067
2024-06-02 13:52:33,350 INFO mapreduce.Job: Job job_1717344892627_0067 running in uber mode : false
2024-06-02 13:52:36,386 INFO mapreduce.Job: map 0% reduce 0%
2024-06-02 13:52:40,412 INFO mapreduce.Job: map 100% reduce 100%
2024-06-02 13:52:41,437 INFO mapreduce.Job: Job job_1717344892627_0067 completed successfully
2024-06-02 13:52:41,496 INFO mapreduce.Job: Counters: 54
File System Counters
FILE: Number of bytes read=701802
FILE: Number of bytes written=1930977
```

Figure 2. Run *Line Count* Program

```

manish-kotra@ROG-M16: ~/Hadoop/LineCount
Combine output records=0
Reduce input groups=1
Reduce shuffle bytes=701002
Reduce input records=58483
Reduce output records=1
Spilled Records=116966
Shuffled Maps =1
Failed Shuffles=0
Merged Map outputs=1
GC time elapsed (ms)=15
CPU time spent (ms)=1540
Physical memory (bytes) snapshot=566177792
Virtual memory (bytes) snapshot=5509476352
Total committed heap usage (bytes)=520093696
Peak Map Physical memory (bytes)=316047360
Peak Map Virtual memory (bytes)=2747224064
Peak Reduce Physical memory (bytes)=250130432
Peak Reduce Virtual memory (bytes)=2762252288

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=2555806
File Output Format Counters
Bytes Written=12
(base) manish-kotra@ROG-M16:~/Hadoop/LineCount$ hadoop dfs -cat /LineCount/Output/*
WARNING: Use of this script to execute dfs is deprecated.
WARNING: Attempting to execute replacement "hdfs dfs" instead.

lines  58483
(base) manish-kotra@ROG-M16:~/Hadoop/LineCount$

```

Figure 3. Output of *Line Count* program

2. The problem setup is given in Figure 4 below. The input commands and the final centroids for k-means algorithm for $k=5$ are shown in Figure 5 and 6, respectively. Similarly, input commands and the final centroids for k-means algorithm for $k=8$ are shown in Figure 7 and 8, respectively.

```

manish-kotra@ROG-M16: ~/Hadoop/Kmeans
(base) manish-kotra@ROG-M16: $ hadoop fs -mkdir /Kmeans
(base) manish-kotra@ROG-M16: $ hadoop fs -mkdir /Kmeans/Input
(base) manish-kotra@ROG-M16: $ hadoop fs -mkdir /Kmeans/Output_k5
(base) manish-kotra@ROG-M16: $ hadoop fs -mkdir /Kmeans/Output_k8
(base) manish-kotra@ROG-M16: $ hadoop fs -put '/home/manish-kotra/Hadoop/Kmeans/input_data/data_points.txt' /
Kmeans/Input
(base) manish-kotra@ROG-M16: $ cd /home/manish-kotra/Hadoop/Kmeans
(base) manish-kotra@ROG-M16:~/Hadoop/Kmeans$ start-dfs.sh
Starting namenodes on [localhost]
localhost: namenode is running as process 313890. Stop it first.
Starting datanodes
localhost: datanode is running as process 314071. Stop it first.
Starting secondary namenodes [ROG-M16]
ROG-M16: secondarynamenode is running as process 314323. Stop it first.
(base) manish-kotra@ROG-M16:~/Hadoop/Kmeans$ start-yarn.sh
Starting resourcemanager
resourcemanager is running as process 314668. Stop it first.
(base) manish-kotra@ROG-M16:~/Hadoop/Kmeans$ jps
335028 Jps
314071 DataNode
313890 NameNode
314323 SecondaryNameNode
314668 ResourceManager
315005 NodeManager
278814 org.eclipse.equinox.launcher_1.6.800.v20240513-1750.jar
(base) manish-kotra@ROG-M16:~/Hadoop/Kmeans$

```

Figure 4. Initialize *KMeans Clustering* program

```

manish-kotra@ROG-M16: ~/Hadoop/Kmeans
(base) manish-kotra@ROG-M16: ~/Hadoop/Kmeans$ export HADOOP_CLASSPATH=$(hadoop classpath)
(base) manish-kotra@ROG-M16: ~/Hadoop/Kmeans$ javac -classpath $(HADOOP_CLASSPATH) -d '/home/manish-kotra/Hadoop/Kmeans/classes_k5' 'KMeans_k5.java'
(base) manish-kotra@ROG-M16: ~/Hadoop/Kmeans$ jar -cvf KMeans_k5.jar -C classes_k5/ .
added manifest
adding: KMeans_k5$PointsReducer.class(in = 3680) (out= 1492)(deflated 59%)
adding: KMeans_k5.class(in = 4209) (out= 1986)(deflated 52%)
adding: KMeans_k5$Point.class(in = 1530) (out= 858)(deflated 43%)
adding: KMeans_k5$PointsMapper.class(in = 3414) (out= 1397)(deflated 59%)
(base) manish-kotra@ROG-M16: ~/Hadoop/Kmeans$ hadoop jar '/home/manish-kotra/Hadoop/Kmeans/KMeans_k5.jar' KMeans_k5 /Kmeans/Input /Kmeans/Output k5/
2024-06-02 13:03:35,683 INFO zlib.ZlibFactory: Successfully loaded & initialized native-zlib library
2024-06-02 13:03:35,683 INFO compress.CodecPool: Got brand-new compressor [.deflate]
2024-06-02 13:03:35,802 INFO client.DefaultHARMAFailoverProxyProvider: Connecting to ResourceManager at /127.0.0.1:8032
2024-06-02 13:03:35,937 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yan/staging/manish-kotra/.staging/job_1717344892627_0022
2024-06-02 13:03:35,979 INFO input.FileInputFormat: Total input files to process : 1
2024-06-02 13:03:36,829 INFO mapreduce.JobSubmitter: number of splits:1
2024-06-02 13:03:37,312 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1717344892627_0022
2024-06-02 13:03:37,312 INFO mapreduce.JobSubmitter: Executing with tokens: []
2024-06-02 13:03:37,395 INFO conf.Configuration: resource-types.xml not found
2024-06-02 13:03:37,395 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2024-06-02 13:03:37,423 INFO impl.YarnClientImpl: Submitted application application_1717344892627_0022
2024-06-02 13:03:37,441 INFO mapreduce.Job: The url to track the job: http://ROG-M16:8088/proxy/application_1717344892627_0022/
2024-06-02 13:03:37,441 INFO mapreduce.Job: Running job: job_1717344892627_0022
2024-06-02 13:03:41,491 INFO mapreduce.Job: Job job_1717344892627_0022 running in uber mode : false
2024-06-02 13:03:41,492 INFO mapreduce.Job: map 0% reduce 0%

```

Figure 5. Run *KMeans Clustering* Program for k=5

```

manish-kotra@ROG-M16: ~/Hadoop/Kmeans
Failed Shuffles=0
Merged Map outputs=1
GC time elapsed (ms)=108
CPU time spent (ms)=7180
Physical memory (bytes) snapshot=822943744
Virtual memory (bytes) snapshot=5534089216
Total committed heap usage (bytes)=525336576
Peak Map Physical memory (bytes)=423223296
Peak Map Virtual memory (bytes)=2771066880
Peak Reduce Physical memory (bytes)=399720448
Peak Reduce Virtual memory (bytes)=2763022336

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=36938010
File Output Format Counters
Bytes Written=187
2024-06-02 13:08:37,475 INFO compress.CodecPool: Got brand-new decompressor [.deflate]
10.135605397809522,21.048029544356027
34.827469318292344,-4.042039644765277
35.14994081310557,4.426826451345206
50.05917926418,30.16527502449059
9.895090760713655,12.564133298076815
(base) manish-kotra@ROG-M16: ~/Hadoop/Kmeans$

```

Figure 6. Output of *KMeans Clustering* program for k=5

```

manish-kotra@ROG-M16: ~/Hadoop/KMeans
(base) manish-kotra@ROG-M16: ~/Hadoop/KMeans$ export HADOOP_CLASSPATH=$(hadoop classpath)
(base) manish-kotra@ROG-M16: ~/Hadoop/KMeans$ javac -classpath $(HADOOP_CLASSPATH) -d '/home/manish-kotra/Hadoop/KMeans/classes_k8' '/home/manish-kotra/Hadoop/KMeans/KMeans_k8.java'
(base) manish-kotra@ROG-M16: ~/Hadoop/KMeans$ jar -cvf KMeans_k8.jar -C classes_k8/ .
added manifest
adding: KMeans_k8$PointsMapper.class(in = 3414) (out= 1396)(deflated 59%)
adding: KMeans_k8.class(in = 4210) (out= 1986)(deflated 52%)
adding: KMeans_k8$Point.class(in = 1530) (out= 859)(deflated 43%)
adding: KMeans_k8$PointsReducer.class(in = 3680) (out= 1491)(deflated 59%)
(base) manish-kotra@ROG-M16: ~/Hadoop/KMeans$ hadoop jar '/home/manish-kotra/Hadoop/KMeans/KMeans_k8.jar' KMeans_k8 /KMeans/Input /KMeans/Output_k8/Output
2024-06-02 13:21:01,580 INFO zlib.ZlibFactory: Successfully loaded & initialized native-zlib library
2024-06-02 13:21:01,580 INFO compress.CodecPool: Got brand-new compressor [.deflate]
2024-06-02 13:21:01,700 INFO client.DefaultHARMFailoverProxyProvider: Connecting to ResourceManager at /127.0.0.1:8032
2024-06-02 13:21:01,838 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-ya
rn/staging/manish-kotra/.staging/job_1717344892627_0037
2024-06-02 13:21:01,883 INFO input.FileInputFormat: Total input files to process : 1
2024-06-02 13:21:01,903 INFO mapreduce.JobSubmitter: number of splits:1
2024-06-02 13:21:01,969 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1717344892627_0037
2024-06-02 13:21:01,969 INFO mapreduce.JobSubmitter: Executing with tokens: []
2024-06-02 13:21:02,048 INFO conf.Configuration: resource-types.xml not found
2024-06-02 13:21:02,048 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2024-06-02 13:21:02,076 INFO impl.YarnClientImpl: Submitted application application_1717344892627_0037
2024-06-02 13:21:02,093 INFO mapreduce.Job: The url to track the job: http://ROG-M16:8088/proxy/application_1717344892627_0037/
2024-06-02 13:21:02,093 INFO mapreduce.Job: Running job: job_1717344892627_0037
2024-06-02 13:21:06,131 INFO mapreduce.Job: Job job_1717344892627_0037 running in uber mode : false
2024-06-02 13:21:06,132 INFO mapreduce.Job: map 0% reduce 0%

```

Figure 7. Run *KMeans Clustering* Program for k=8

```

manish-kotra@ROG-M16: ~/Hadoop/KMeans
CPU time spent (ns)=6640
Physical memory (bytes) snapshot=835760128
Virtual memory (bytes) snapshot=534744576
Total committed heap usage (bytes)=525336576
Peak Map Physical memory (bytes)=433262592
Peak Map Virtual memory (bytes)=2764955648
Peak Reduce Physical memory (bytes)=402497536
Peak Reduce Virtual memory (bytes)=2769788928
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=36938010
File Output Format Counters
Bytes Written=307
2024-06-02 13:25:59,477 INFO compress.CodecPool: Got brand-new decompressor [.deflate]
10.047880963993261,22.040376261194265
16.9551981281736,8.538945427486874
34.70378363921685,-1.0893446502684688
35.790900645301434,-7.0352605722674035
37.672419320399655,10.691614004078899
5.493841574650159,4.182395873792595
50.154045227868046,30.363434484221877
9.726316384154488,14.347276674397174
(base) manish-kotra@ROG-M16: ~/Hadoop/KMeans$

```

Figure 8. Output of *KMeans Clustering* program for k=8

3. Advantages of Using K-Means Clustering with MapReduce:

- **Scalability:** MapReduce allows K-Means to handle very large datasets by distributing the computation across multiple nodes, making it scalable to big data scenarios.
- **Parallel Processing:** The MapReduce framework inherently supports parallel processing, which can significantly speed up the clustering process.
- **Efficiency:** By breaking down the K-Means algorithm into map and reduce tasks, it can efficiently process large amounts of data in parallel, reducing the overall computation time.

Disadvantages of Using K-Means Clustering with MapReduce:

- Complexity: Implementing K-Means with MapReduce adds complexity to the setup and configuration, requiring knowledge of both the clustering algorithm and the MapReduce framework.
 - Communication Overhead: There can be significant communication overhead between the map and reduce phases, which can impact performance, especially if the dataset is not large enough to justify the overhead.
4. Yes, the number of distance comparisons can be reduced by applying the Canopy Selection technique. This approach uses a cheap, approximate distance measure to group data points into overlapping subsets called canopies. Only the pairs of points within the same canopy are then considered for exact distance measurements, significantly reducing the number of expensive distance computations needed.

The distance metric used for canopy clustering should be one that is computationally inexpensive and able to provide a rough estimate of similarity. A commonly used metric is the inverted index, commonly used in information retrieval systems, are very efficient in high dimensions and can find elements near the query by examining only a small fraction of a data set.

5. Yes, it is possible to apply Canopy Selection on MapReduce. Here's a high-level implementation approach:
- Map Phase: Each data point is assigned to multiple mappers based on a cheap distance metric. Each mapper processes a subset of data points and identifies potential canopies by assigning points to canopies if they fall within a loose threshold distance from a randomly selected center point. The mapper outputs canopy assignments for each data point.
 - Reduce Phase: Reducers collect canopy assignments from mapper and refine the canopies by collecting all the points belonging to the same canopy. Reducers then finalize the canopies, outputting the data points and their associated canopies.
6. Yes, it is possible to combine Canopy Selection with K-Means on MapReduce. The process involves the first two steps given in answer to question 5 above. After applying these steps of canopy selection the following steps of K-means algorithms is used to refine clusters and centroids:
- K-Means Initialization (Map Phase): Within each canopy, mappers initialize K-Means by selecting initial centroids. Each mapper processes points in its canopy, computing initial distances to centroids.
 - K-Means Iteration (Map and Reduce Phases): Map Phase: Mappers assign points to the nearest centroid within each canopy. Reduce Phase: Reducers recompute the centroids based on the assigned points. These steps are iterated until convergence.