**Distributed Systems**

**Project 2B: Harp Kmeans**

**Steps and Data Flow in Harp Kmeans: -**

1. KmeansMapCollective.java

This is the main class where the execution of harp kmean starts. The first step is to parse the arguments accordingly. Then, the next step is to generate Datapoints based on the arguments passed.

To generate Data points we use two values (numofDatapoints and vector size). We use the build in java function to generate random Double values and based on the size of vector, we create a vector of random double values and assign them accordingly.

Next step is to create Initial Centroids. We assign initial centroid values for the algorithm to start its initial steps.

Then, we configure the parameter settings for the Hadoop MR job by creating a job instance and using jobConfig. The next step is the actual Job execution.

1. KmeansMapper.java

This is the class where the Harp Kmeans execution starts. We initially define a method to setup values for the job specifications.

Then based on the number of iterations mentioned, we execute the algorithm where each data point is assigned to a centroid based on the equilibrium distance calculated. These centroids are local to the execution. They are received from the master using broadcastCentroids() method.

Then, once all the child processes are completed, we calculate the overall centroids by aggregating the values through allreduce() method. This will give us the global centroids and their values. These values are then written to a file in HDFS using the outputCentroids() method.

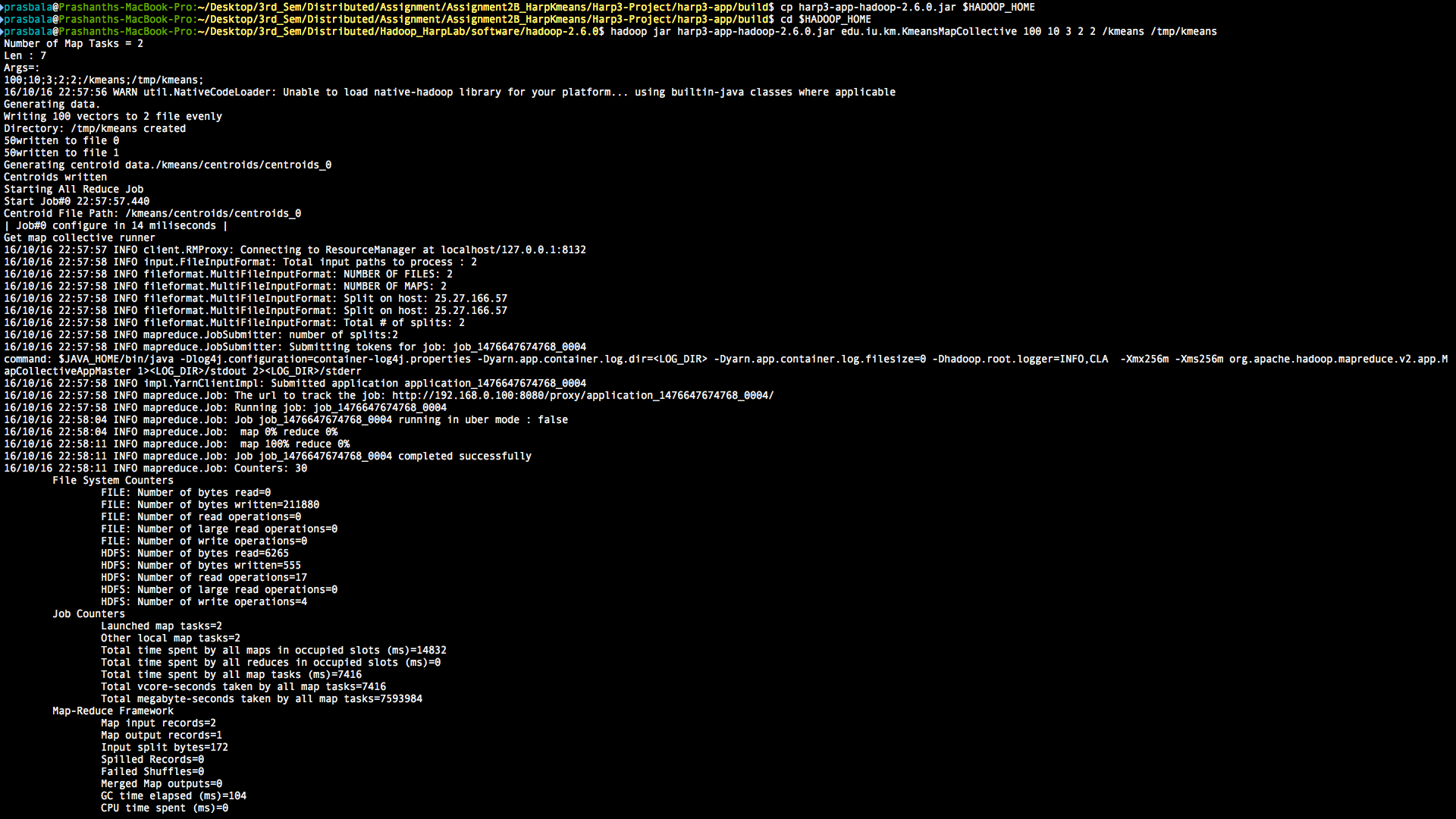
1. KMeansConstants.java

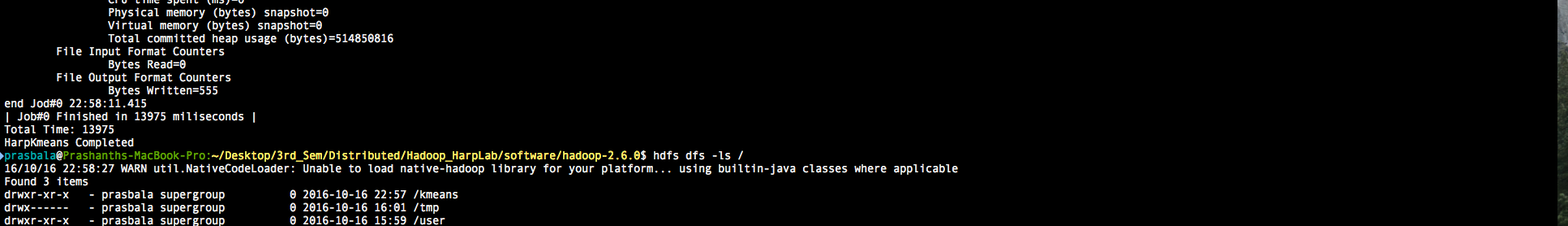
This is the class which contains any constant values which are being used in the main classes.

1. Utils.java

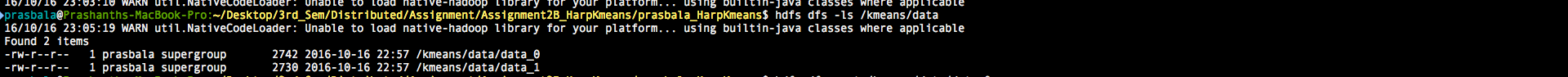
This class file contains generateData() method, generateInitialCentroids() method and loadData() method. These methods are basically used to generate and load the input data into HDFS.

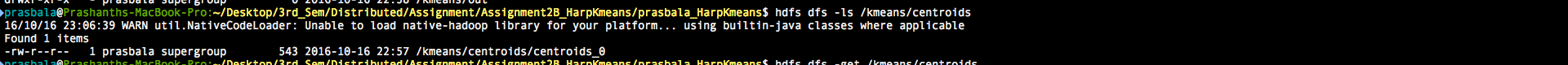
**Harp Kmeans Execution: -**

****

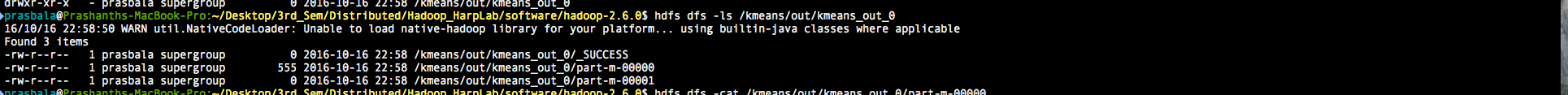
****

**Data Files and Initial Centroids in HDFS: -**

****

****

**Output Files generated in HDFS: -**

****

**I have attached the Data files and output files from the above execution separately in the project folder.**