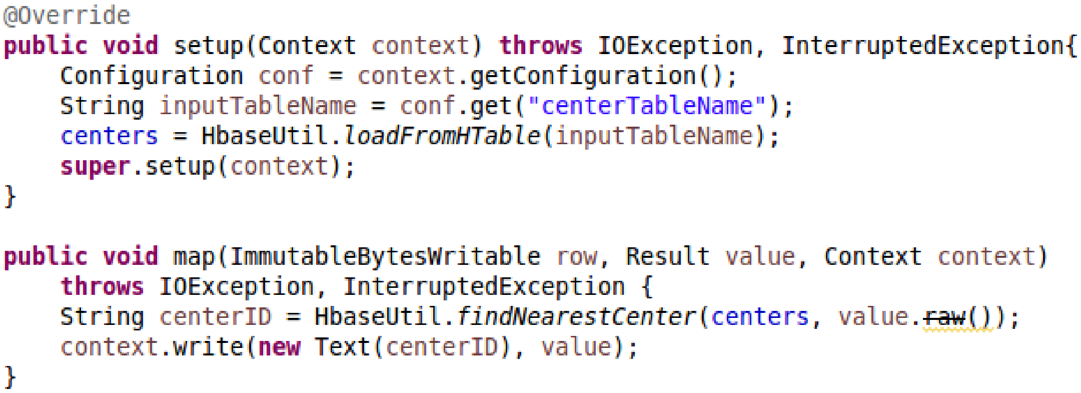
K-Means Clustering on Energy Efficiency Data Set.

Abhishek Madav # 86378148

The program for the K-Means clustering uses HBase as its primary data source. The program accepts four parameters of the form *"<input file> <input table name> <center table name> <K points>".*

First job (dataLoadJob) for Loading Data inputs values from the dataset.txt file loaded in the HDFS. The values string is split using delimiters and stored into a double array and put into the "*input table name"* table. The main program waits until the mapper is done with the data loading. No reducer is required for this job since the task is accomplished by the mapper alone.

Second job (clusterJob) handles the clustering task over the dataset stored into the HBase table. The mapper sets the current view of the centers in the setup(). Each value from the HBase input table gets treated as a single point and the relative distance of this point with all the current centers is calculated with the minimum of them being sent as the selected center. The mapper emits a <*key, value*> as <*nearest center ID, double values of a single row in HBase table*>



```
@Override
public void setup(Context context) throws IOException, InterruptedException{
    Configuration conf = context.getConfiguration();
    String inputTableName = conf.get("centerTableName");
    centers = HbaseUtil.loadFromHTable(inputTableName);
    super.setup(context);
}

public void map(ImmutableBytesWritable row, Result value, Context context)
    throws IOException, InterruptedException {
    String centerID = HbaseUtil.findNearestCenter(centers, value.~~raw~~());
    context.write(new Text(centerID), value);
}
```

K number of Reducers for the job, see the nearest center value for each HBase entry. The reducer finds the average of the center values and checks the distance between the averaged center and the old set of the centers. If the distance is greater than 0.1, the program increments a global counter maintained for the iterative map-reduce for the K-means clustering. The program iteratively executes on finding this incremented value of the counter. Each time the reducer sees the distance greater than the threshold, it updates the center table to reflect this latest value. The center table finds the most updated value after few iterations.