<u>CSE 512 DDS Project - Phase 1</u>

Group Members – Group 2

- Kunwar Chowhan (1213420154)
- Aditya Chayapathy (1213050538)
- Kevin Thomas (1213122571)

YouTube Link: https://youtu.be/rph6LAWIn6w

System Description:

The following Google Cloud Engine n1-standard-1 (1vCPU , 3.75GB) instances were used for the purpose of demonstration:

- Master – 10.142.0.2
- Slave 1 – 10.142.0.3
- Slave 2 – 10.142.0.4

Commands to Load Data to HDFS:

- hadoop fs -put /home/hduser/Downloads/zcta510.csv /user/kunwar/
- hadoop fs -put /home/hduser/Downloads/arealm.csv /user/kunwar/

Queries:

```
// Problem 1 - Create PointRDD
import org.datasyslab.geospark.spatialRDD.PointRDD
import org.datasyslab.geospark.enums.FileDataSplitter
import org.apache.spark.storage.StorageLevel

val points="hdfs://master:54310/user/kunwar/arealm.csv"
val pointRDD = new PointRDD(sc, points, 0, FileDataSplitter.CSV, false,StorageLevel.MEMORY_ONLY);


//Problem 2a – Range Query without R-Tree Index
import org.datasyslab.geospark.spatialOperator.RangeQuery;
import org.datasyslab.geospark.spatialRDD.PointRDD;
import com.vividsolutions.jts.geom.Envelope;
import org.datasyslab.geospark.enums.FileDataSplitter;

val points="hdfs://master:54310/user/kunwar/arealm.csv"
val queryEnvelope=new Envelope (-113.79,-109.73,32.99,35.08);
val objectRDD = new PointRDD(sc, points, 0, FileDataSplitter.CSV, false, StorageLevel.MEMORY_ONLY);
val resultSize = RangeQuery.SpatialRangeQuery(objectRDD, queryEnvelope, false, false).count();
```

//Problem 2b – Range Query with R-Tree Index

```
import org.datasyslab.geospark.spatialOperator.RangeQuery;
import org.datasyslab.geospark.spatialRDD.PointRDD;
import com.vividsolutions.jts.geom.Envelope;
import org.datasyslab.geospark.enums.FileDataSplitter;
import org.datasyslab.geospark.enums.IndexType;

val points="hdfs://master:54310/user/kunwar/arealm.csv"
val queryEnvelope=new Envelope (-113.79,-109.73,32.99,35.08);
val objectRDD = new PointRDD(sc, points, 0, FileDataSplitter.CSV, false, StorageLevel.MEMORY_ONLY);
objectRDD.buildIndex(IndexType.RTREE,false);
val resultSize = RangeQuery.SpatialRangeQuery(objectRDD, queryEnvelope, false, true).count();
```

//Problem 3a – kNN without R-Tree Index

```
import org.datasyslab.geospark.spatialOperator.KNNQuery;
import org.datasyslab.geospark.spatialRDD.PointRDD;
import com.vividsolutions.jts.geom.GeometryFactory;
import com.vividsolutions.jts.geom.Point;
import com.vividsolutions.jts.geom.Coordinate;
import org.datasyslab.geospark.enums.FileDataSplitter;

val points="hdfs://master:54310/user/kunwar/arealm.csv"
val fact=new GeometryFactory();
val queryPoint=fact.createPoint(new Coordinate(35.08,-113.79));
val objectRDD = new PointRDD(sc, points, 0, FileDataSplitter.CSV, false, StorageLevel.MEMORY_ONLY);
val resultSize = KNNQuery.SpatialKnnQuery(objectRDD, queryPoint, 5,false).size();
```

//Problem 3b - kNN with R-Tree Index

```
import org.datasyslab.geospark.spatialOperator.KNNQuery;
import org.datasyslab.geospark.spatialRDD.PointRDD;
import com.vividsolutions.jts.geom.GeometryFactory;
import com.vividsolutions.jts.geom.Point;
import com.vividsolutions.jts.geom.Coordinate;
import org.datasyslab.geospark.enums.FileDataSplitter;
import org.datasyslab.geospark.enums.IndexType;

val points="hdfs://master:54310/user/kunwar/arealm.csv"
val fact=new GeometryFactory();
val queryPoint=fact.createPoint(new Coordinate(35.08,-113.79));
val objectRDD = new PointRDD(sc, points, 0, FileDataSplitter.CSV, false, StorageLevel.MEMORY_ONLY);
objectRDD.buildIndex(IndexType.RTREE,false);
```

val resultSize = KNNQuery.SpatialKnnQuery(objectRDD, queryPoint, 5,true).size();

//Problem 4a - Join without R-Tree index

```
import org.datasyslab.geospark.spatialOperator.JoinQuery;
import org.datasyslab.geospark.spatialRDD.PointRDD;
import org.datasyslab.geospark.spatialRDD.RectangleRDD;
import org.datasyslab.geospark.enums.FileDataSplitter;
import org.datasyslab.geospark.enums.GridType;

val points="hdfs://master:54310/user/kunwar/arealm.csv"
val rectangle="hdfs://master:54310/user/kunwar/zcta510.csv"
val objectRDD = new PointRDD(sc, points, 0, FileDataSplitter.CSV, false, StorageLevel.MEMORY_ONLY);
val rectangleRDD = new RectangleRDD(sc, rectangle, 0, FileDataSplitter.CSV, false,
StorageLevel.MEMORY_ONLY);
objectRDD.spatialPartitioning(GridType.EQUALGRID);
rectangleRDD.spatialPartitioning(objectRDD.grids);
val resultSize = JoinQuery.SpatialJoinQuery
(objectRDD,rectangleRDD,false,false).count();
```

//Problem 4b -0 Join with R-Tree index

```
import org.datasyslab.geospark.spatialOperator.JoinQuery;
import org.datasyslab.geospark.spatialRDD.PointRDD;
import org.datasyslab.geospark.spatialRDD.RectangleRDD;
import org.datasyslab.geospark.enums.FileDataSplitter;
import org.datasyslab.geospark.enums.GridType;
import org.datasyslab.geospark.enums.IndexType;

val points="hdfs://master:54310/user/kunwar/arealm.csv"
val rectangle="hdfs://master:54310/user/kunwar/zcta510.csv"
val objectRDD = new PointRDD(sc, points, 0, FileDataSplitter.CSV, false, StorageLevel.MEMORY_ONLY);
val rectangleRDD = new RectangleRDD(sc, rectangle, 0, FileDataSplitter.CSV, false);
objectRDD.spatialPartitioning(GridType.EQUALGRID);
objectRDD.buildIndex(IndexType.RTREE,true);
rectangleRDD.spatialPartitioning(objectRDD.grids);
val resultSize = JoinQuery.SpatialJoinQuery(objectRDD,rectangleRDD,true, false).count();
```

// Problem 4c - Join with R-tree grid without R-tree index

```
import org.datasyslab.geospark.spatialOperator.JoinQuery;
import org.datasyslab.geospark.spatialRDD.PointRDD;
import org.datasyslab.geospark.spatialRDD.RectangleRDD;
import org.datasyslab.geospark.enums.FileDataSplitter;
import org.datasyslab.geospark.enums.GridType;

val points="hdfs://master:54310/user/kunwar/arealm.csv"
val rectangle="hdfs://master:54310/user/kunwar/zcta510.csv"
val objectRDD = new PointRDD(sc, points, 0, FileDataSplitter.CSV, false, StorageLevel.MEMORY_ONLY);
val rectangleRDD = new RectangleRDD(sc, rectangle, 0, FileDataSplitter.CSV, false,
StorageLevel.MEMORY_ONLY);
```

```
objectRDD.spatialPartitioning(GridType.RTREE);
rectangleRDD.spatialPartitioning(objectRDD.grids);
val resultSize = JoinQuery.SpatialJoinQuery(objectRDD,rectangleRDD,false,false).count();
```