BIG DATA ANALYTICS DA 1 CODE TOPIC: HEART RATE (BPM)

MapReduce Code in Java

The code consists of three parts: the Mapper, Reducer, and Driver classes.

1. HeartRateMapper.java

The Mapper class reads the input data, extracts the heart rate and activity level, and emits them as key-value pairs.

import org.apache.hadoop.io.IntWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapreduce.Mapper;

import java.io.IOException;

public class HeartRateMapper extends Mapper<Object, Text, Text, IntWritable> {

private Text activityLevel = new Text();

private IntWritable heartRate = new IntWritable();

@Override

protected void map(Object key, Text value, Context context) throws IOException, InterruptedException {

String[] parts = value.toString().split(",");

if (!parts[0].equals("Timestamp")) {

String activity = parts[2].trim();

int rate = Integer.parseInt(parts[1].trim());

activityLevel.set(activity);

heartRate.set(rate);

context.write(activityLevel, heartRate);

}

}

}

2. HeartRateReducer.java

The Reducer class aggregates heart rate values by activity level and computes the average heart rate for each.

import org.apache.hadoop.io.IntWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapreduce.Reducer;

import java.io.IOException;

public class HeartRateReducer extends Reducer<Text, IntWritable, Text, IntWritable> {

private IntWritable averageHeartRate = new IntWritable();

@Override

protected void reduce(Text key, Iterable<IntWritable> values, Context context) throws IOException, InterruptedException {

int sum = 0;

int count = 0;

for (IntWritable value : values) {

sum += value.get();

count++;

}

int average = count == 0 ? 0 : sum / count;

averageHeartRate.set(average);

context.write(key, averageHeartRate);

}

}

3. HeartRateDriver.java

The Driver class configures and runs the MapReduce job.

import org.apache.hadoop.conf.Configuration;

import org.apache.hadoop.fs.Path;

import org.apache.hadoop.io.IntWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapreduce.Job;

import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;

import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

public class HeartRateDriver {

public static void main(String[] args) throws Exception {

if (args.length != 2) {

System.err.println("Usage: HeartRateDriver <input path> <output path>");

System.exit(-1);

}

Configuration conf = new Configuration();

Job job = Job.getInstance(conf, "Heart Rate Analysis");

job.setJarByClass(HeartRateDriver.class);

job.setMapperClass(HeartRateMapper.class);

job.setReducerClass(HeartRateReducer.class);

job.setOutputKeyClass(Text.class);

job.setOutputValueClass(IntWritable.class);

FileInputFormat.addInputPath(job, new Path(args[0]));

FileOutputFormat.setOutputPath(job, new Path(args[1]));

System.exit(job.waitForCompletion(true) ? 0 : 1);

}

}