**StringSreaching Program:**

package stringSearchJob;

import java.io.IOException;

import java.util.Scanner;

import org.apache.hadoop.conf.Configuration;

import org.apache.hadoop.fs.Path;

import org.apache.hadoop.io.IntWritable;

import org.apache.hadoop.io.LongWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapred.JobClient;

import org.apache.hadoop.mapred.JobConf;

import org.apache.hadoop.mapreduce.Job;

import org.apache.hadoop.mapreduce.Mapper;

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

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

public class StringSearch{

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

try {

if (argv.length<3) {

System.err.println("Give the input/ output/ keyword!");

return;

}

JobConf conf = new JobConf(StringSearch.class);

Job job = new Job(conf,"StringSearch");

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

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

conf.set("search", argv[2]);

job.setJarByClass(StringSearch.class);

job.setMapperClass(WordMapper.class);

job.setNumReduceTasks(0);

job.setMapOutputKeyClass(Text.class);

job.setMapOutputValueClass(IntWritable.class);

job.setOutputKeyClass(Text.class);

job.setOutputValueClass(IntWritable.class);

JobClient.runJob(conf);

job.waitForCompletion(true);

}

catch (Exception e) {

e.printStackTrace();

}

}

public static class WordMapper extends Mapper<LongWritable, Text, Text, IntWritable>{

@Override

public void map(LongWritable key, Text value, Context context) throws IOException, InterruptedException {

try {

Configuration conf = context.getConfiguration();

String search = conf.get("search");

String line = value.toString();

Scanner scanner = new Scanner(line);

while (scanner.hasNext()) {

if (line.contains(search)) {

String line1 = scanner.next();

context.write(new Text(line1), new IntWritable(1));

}

}

scanner.close();

}

catch (IOException e){

e.printStackTrace();

}

catch (InterruptedException e){

e.printStackTrace();

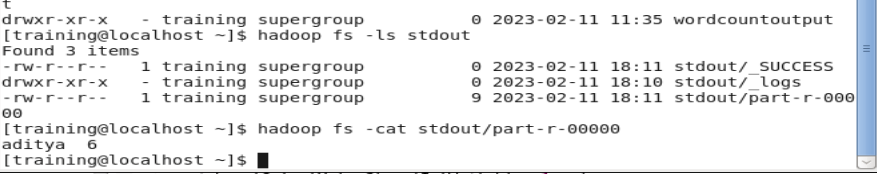
}

}

}

}

**Output:**

****