**//example took help from inter net and “Hadoop in practice”**

**package** filesamples;

**import** java.io.IOException;

**import** java.util.Arrays;

**import** org.apache.hadoop.io.LongWritable;

**import** org.apache.hadoop.io.Text;

**import** org.apache.hadoop.mapred.MapReduceBase;

**import** org.apache.hadoop.mapred.Mapper;

**import** org.apache.hadoop.mapred.OutputCollector;

**import** org.apache.hadoop.mapred.Reporter;

/\*\*

\* The Anagram mapper class gets a word as a line from the HDFS input and sorts the

\* letters in the word and writes its back to the output collector as

\* Key : sorted word (letters in the word sorted)

\* Value: the word itself as the value.

\* When the reducer runs then we can group anagrams togather based on the sorted key.

\*

\* **@author** Harjinder Singh

\*

\*/

**public** **class** Amapper **extends** MapReduceBase **implements**

Mapper<LongWritable, Text, Text, Text> {

**private** Text sortedText = **new** Text();

**private** Text orginalText = **new** Text();

**public** **void** map(LongWritable key, Text value,

OutputCollector<Text, Text> outputCollector, Reporter reporter)

**throws** IOException {

String word = value.toString();

**char**[] wordChars = word.toCharArray();

Arrays.*sort*(wordChars);

String sortedWord = **new** String(wordChars);

sortedText.set(sortedWord);

orginalText.set(word);

outputCollector.collect(sortedText, orginalText);

}

}

////////////////////////////////////////////////////////////////////////////

**package** filesamples;

**import** java.io.IOException;

**import** java.util.Iterator;

**import** java.util.StringTokenizer;

**import** org.apache.hadoop.io.Text;

**import** org.apache.hadoop.mapred.MapReduceBase;

**import** org.apache.hadoop.mapred.OutputCollector;

**import** org.apache.hadoop.mapred.Reducer;

**import** org.apache.hadoop.mapred.Reporter;

/\*\*

\* The Anagram reducer class groups the values of the sorted keys that came in and

\* checks to see if the values iterator contains more than one word. if the values

\* contain more than one word we have spotted a anagram.

\* **@author** harjinder

\*

\*/

**public** **class** Areducer **extends** MapReduceBase **implements** Reducer<Text, Text, Text, Text> {

**private** Text outputKey = **new** Text();

**private** Text outputValue = **new** Text();

**public** **void** reduce(Text aKey, Iterator<Text> aValues,

OutputCollector<Text, Text> results, Reporter reporter) **throws** IOException {

String output = "";

**while**(aValues.hasNext())

{

Text anagam = aValues.next();

output = output + anagam.toString() + "~";

}

StringTokenizer outputTokenizer = **new** StringTokenizer(output,"~");

**if**(outputTokenizer.countTokens()>=2)

{

output = output.replace("~", "//");

outputKey.set(aKey.toString());

outputValue.set(output);

results.collect(outputKey, outputValue);

}

}

}

/////////////////////////////////////////////////////////////////////////////////////

**package** filesamples;

**import** org.apache.hadoop.fs.Path;

**import** org.apache.hadoop.io.Text;

**import** org.apache.hadoop.mapred.FileInputFormat;

**import** org.apache.hadoop.mapred.FileOutputFormat;

**import** org.apache.hadoop.mapred.JobClient;

**import** org.apache.hadoop.mapred.JobConf;

**import** org.apache.hadoop.mapred.TextInputFormat;

**import** org.apache.hadoop.mapred.TextOutputFormat;

**public** **class** Acount {

/\*\*

\* **@param** args

\*/

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

JobConf conf = **new** JobConf(Acount.**class**);

conf.setJobName("anagramcount");

conf.setOutputKeyClass(Text.**class**);

conf.setOutputValueClass(Text.**class**);

conf.setMapperClass(Amapper.**class**);

// conf.setCombinerClass(AnagramReducer.class);

conf.setReducerClass(Areducer.**class**);

conf.setInputFormat(TextInputFormat.**class**);

conf.setOutputFormat(TextOutputFormat.**class**);

FileInputFormat.*setInputPaths*(conf, **new** Path("input/txtSample"));

FileOutputFormat.*setOutputPath*(conf, **new** Path("out"));

JobClient.*runJob*(conf);

}

}