**Map Reduce Word Count Problem :**

**STEP 1:**

Create the text File mytext.txt in the local file system /home/cloudera

$ Vi mytext.txt

Press i to enable enter text into mytext

Hello all welcome to the first program

Of Hadoop word count

This program of Hadoop is to count the occurrences

Of word in the text file

Save this -> Esc ->:!wq

OR

We can use any editor to create the above text file in the same location

**STEP 2: Create directory into Hadoop**

**$hadoop fs –mkdir /user/input**

**STEP 3: Move mytext file into Hadoop directory**

Open the terminal

$hadoop fs –put /home/cloudera/mytext.txt /user/input

**STEP4: Open the eclipse to write map reduce program in Java**

Goto File-> New->Java Project -> MapReduceP (give this name as a project)

Goto Navigation Pan of project ->src->right click -> New->Package->mapreduce(this is the package name)

Goto Navigation Pan of project ->src->right click -> New->class->WordCount(this is the class name)

Write the below mentioned program :

import java.io.IOException;

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.mapreduce.Job;

import org.apache.hadoop.mapreduce.Mapper;

import org.apache.hadoop.mapreduce.Reducer;

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

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

public class WordCount {

// Map function

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

private Text word = new Text();

public void map(LongWritable key, Text value, Context context)

throws IOException, InterruptedException {

// Splitting the line on spaces

String[] stringArr = value.toString().split("\\s+");

for (String str : stringArr) {

word.set(str);

context.write(word, new IntWritable(1));

}

}

}

// Reduce function

public static class MyReducer extends Reducer<Text, IntWritable, Text, IntWritable>{

private IntWritable result = new IntWritable();

public void reduce(Text key, Iterable<IntWritable> values, Context context)

throws IOException, InterruptedException {

int sum = 0;

for (IntWritable val : values) {

sum += val.get();

}

result.set(sum);

context.write(key, result);

}

}

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

Configuration conf = new Configuration();

Job job = Job.getInstance(conf, "WC");

job.setJarByClass(WordCount.class);

job.setMapperClass(MyMapper.class);

job.setReducerClass(MyReducer.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);

}

}

**STEP 5: Add the Hadoop Jars**

Right click on project name ->build path -> Add External Jars

External jars of Hadoop are available

User->lib->Hadoop->client

User->lib->Hadoop->client.2.0

User->lib->Hadoop->lib

Add all the jars

**STEP6: Create the Jar file**

Right click on program ->export->Java->Jars->next->choose the location where to store jar

**STEP7: Get the location of all necessary files to execute the mapreduce wordcount problem**

**Jar file** is at /home/cloudera/test.jar

**class file** is at /home/cloudera/workspace/MapReduce/bin/mapreduce/WordCount

**Input file** mytext.txt is at /user/process/mytext.txt

**Output file** location is /user/out

**STEP8: We should run this mapreduce program in following way:**

**$hadoop jar /home/cloudera/bigdata/text.jar mapreduce.WordCount /user/process /user/out**

Output file

Input file

Class file

package

Jar

**STEP 9: open the out put file**

After succesfule execution of this program two files will be created

Success & part-00001

We can check it out

$hadoop fs –ls /user/out

To see the output

$ Hadoop fs –cat /user/out/part-0001

**Note :**

**Don’t worry if get following error after executing the program in eclipse**

WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable

Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: 0

at mapreduce.WordCount.main(WordCount.java:52)

This is because in eclipse we are not supplying arguments if you want to do so you can configure this.